



Technische Universität München  
Fakultät für Informatik  
Fachgebiet Didaktik der Informatik

# Investigating Knowledge Structures in Computer Science Education

Andreas Michael Mühling

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktors der Naturwissenschaften (Dr. rer. nat.)**

genehmigten Dissertation.

Vorsitzende:

Univ.-Prof. Gudrun J. Klinker, Ph.D.

Prüfer:

1. Univ.-Prof. Dr. Peter Hubwieser
2. Prof. Christopher Hundhausen, Ph.D., Washington State University, USA
3. Univ.-Prof. Dr. Johannes Magenheimer, Universität Paderborn

Die Dissertation wurde am 19. Februar 2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 5. Mai 2014 angenommen.



# Abstract

To be knowledgeable in a subject matter is a prerequisite of being competent. Experts typically possess densely connected structural knowledge of the concepts in their field of expertise. Therefore, investigating the knowledge structures of learners remains a central aspect of educational research in computer science as well as in other subjects, despite the ongoing trend towards outcome and competence oriented assessment.

This thesis presents *concept landscapes* - a novel way of investigating the state and development of knowledge structures using concept maps. Instead of focusing on the assessment and evaluation of single maps, the data of many persons is aggregated and data mining approaches are used in analysis. New insights into the “shared” knowledge of groups of learners are possible in this way. The educational theories underlying the approach, the definition of concept landscapes, and the accompanying analysis methods are presented.

Since both data mining techniques and concept map collection are well suited for computer-based approaches, three software projects have been realized in the course of this thesis. They allow computer-based drawing and assessing concept maps, the subsequent analysis of concept maps and concept landscapes, and the extraction of salient concepts and propositions from texts.

The methods and tools have been applied in three research projects that investigate structural knowledge in computer science education. Among others, the structural knowledge of CS students entering university has been analyzed. It has been found, that attending the compulsory school subject “Informatics” in Bavaria has visible effects on the knowledge structures of beginning students. The results of these studies can be taken in order to better understand the possibilities, limits, and effects of teaching methods, curricula, or learning materials in computer science education and other subjects.



# Zusammenfassung

Wissen ist eine Voraussetzung um in einem Fach kompetent zu sein. Experten besitzen üblicherweise eine dicht vernetzte Wissensstruktur hinsichtlich der Konzepte ihres Spezialgebiets. Daher ist die Untersuchung von Wissensstrukturen von Lernenden ein zentraler Aspekt von Lehr-/Lernforschung in Informatik und anderen Fächern, trotz eines erkennbaren Trends hin zur Outcome- und Kompetenzorientierung.

Diese Arbeit präsentiert *concept landscapes* - ein neuartiger Weg um den Zustand und die Entwicklung von Wissensstrukturen mit Hilfe von Begriffsnetzen zu untersuchen. Anstelle den Fokus bei der Bewertung und Auswertung auf einzelne Netz zu richten, werden die Daten von vielen Personen aggregiert und zur Analyse Methoden des Data Mining angewendet. Dadurch sind neue Einblicke in das "gemeinsame" Wissen einer Gruppe von Lernenden möglich. Die Arbeit präsentiert die zugrundeliegenden Theorien der Lehr-/Lernforschung, die Definition von *concept landscapes* und die möglichen Analysemethoden.

Da sich Techniken des Data Mining und die Erhebung von Begriffsnetzen inhärent für rechnerbasierte Verfahren eignen, sind drei Softwareprojekte im Laufe dieser Arbeit realisiert worden. Diese erlauben das rechnerbasierte Zeichnen und Bewerten von Begriffsnetzen, die darauf folgende Analyse von Begriffsnetzen und *concept landscapes* sowieso das Extrahieren von zentralen Konzepten und Propositionen aus Texten.

Die Methoden und Werkzeuge wurden in drei Forschungsprojekten angewendet, die Wissensstrukturen im Bereich der Informatikausbildung untersuchen. Unter anderem wurde das strukturelle Wissen von Informatik-Studienanfängern analysiert. Dabei hat sich gezeigt, dass der Besuch des Pflichtfachs Informatik in Bayern sichtbaren Einfluss auf die Wissensstrukturen von Studienanfängern hat. Die Ergebnisse der Studien helfen, die Möglichkeiten, Grenzen und den Einfluss von Lehrmethoden, Lehrplänen und Lehrmaterialien in der Ausbildung in Informatik und anderen Fächern besser zu verstehen.



# Contents

<b>I Introduction</b>	<b>1</b>
1 Problem Setting	3
2 Detailed Overview	11
<b>II Theoretical Background and Related Work</b>	<b>17</b>
<b>3 Knowledge and Learning</b>	<b>19</b>
3.1 Knowledge . . . . .	19
3.1.1 Psychological Foundations . . . . .	20
3.1.2 Epistemological Foundations . . . . .	24
3.2 Learning . . . . .	25
3.2.1 Psychological Foundations . . . . .	26
3.2.2 Constructivism . . . . .	27
3.2.3 Conceptual Change . . . . .	30
3.2.4 Meaningful Learning . . . . .	33
3.2.5 Models of Learning . . . . .	36
3.3 Assessment . . . . .	37
3.3.1 Learning Objectives . . . . .	39
3.3.1.1 Taxonomies . . . . .	39
3.3.2 Learning Outcomes . . . . .	41
3.3.2.1 SOLO Taxonomy . . . . .	41
3.3.2.2 Competencies . . . . .	42

<b>4</b>	<b>Concept Maps</b>	<b>45</b>
4.1	Elements . . . . .	46
4.1.1	Map Structure . . . . .	50
4.2	Applications . . . . .	52
4.2.1	Learning and Teaching . . . . .	54
4.2.2	Assessment . . . . .	56
4.2.2.1	Scoring System . . . . .	60
<b>5</b>	<b>Analysis Methods</b>	<b>67</b>
5.1	Graph Theory . . . . .	67
5.2	Pathfinder Networks . . . . .	68
5.2.1	Construction . . . . .	69
5.2.2	Investigating Structural Knowledge . . . . .	71
5.3	Cluster Analysis . . . . .	75
5.3.1	Partitioning Methods . . . . .	75
5.3.2	Model Based Methods . . . . .	77
5.4	Text Mining . . . . .	78
5.4.1	Existing Software Solutions . . . . .	80
<b>III</b>	<b>From Concept Maps to Concept Landscapes</b>	<b>83</b>
<b>6</b>	<b>Possibilities and Limitations of Concept Maps</b>	<b>85</b>
6.1	Cognitive View . . . . .	85
6.1.1	Observing the Externalization . . . . .	92
6.2	Epistemological View . . . . .	97
6.2.1	Concept Maps as Graphs . . . . .	98
6.2.1.1	Representations in Computers . . . . .	101
6.3	Educational View . . . . .	103



6.3.1	Assessing Learning . . . . .	104
6.3.2	Scoring Concept Maps . . . . .	108
<b>7</b>	<b>Concept Landscapes</b>	<b>111</b>
7.1	Definition . . . . .	112
7.1.1	Formal Definition . . . . .	117
7.1.1.1	Vertical . . . . .	117
7.1.1.2	Horizontal . . . . .	118
7.2	Analysis Methods . . . . .	119
7.2.1	Cluster Analysis . . . . .	119
7.2.1.1	Similarity Based Clustering . . . . .	122
7.2.1.2	Latent Class Clustering . . . . .	123
7.2.1.3	Example . . . . .	125
7.2.2	Pathfinder . . . . .	131
7.2.2.1	Example . . . . .	135
7.2.3	Graph Measures . . . . .	138
7.2.3.1	Simple Graph Measures . . . . .	138
7.2.3.2	Advanced Graph Measures . . . . .	139
7.2.3.3	Community Detection . . . . .	141
7.2.3.4	Frequent Subgraph Analysis . . . . .	142
7.2.4	Visualization . . . . .	143
7.2.4.1	Vertical Landscapes . . . . .	144
7.2.4.2	Horizontal Landscapes . . . . .	145
<b>8</b>	<b>Software Support for Concept Landscapes</b>	<b>149</b>
8.1	CoMapEd . . . . .	150
8.1.1	Requirements Analysis . . . . .	150
8.1.2	Design and Implementation . . . . .	152

8.2	CoMaTo . . . . .	155
8.2.1	Requirements Analysis . . . . .	155
8.2.2	Design and Implementation . . . . .	156
8.3	ConEx . . . . .	156
8.3.1	Requirements Analysis . . . . .	157
8.3.2	Design and Implementation . . . . .	157
<b>IV</b>	<b>Case Studies</b>	<b>161</b>
<b>9</b>	<b>Overview</b>	<b>163</b>
<b>10</b>	<b>CS1: Computer Science Education for Non-Majors</b>	<b>167</b>
10.1	Description of the Setting . . . . .	167
10.2	Data Collection & Research Questions . . . . .	169
10.3	Analysis and Results . . . . .	173
10.3.1	RQ1: Development of Knowledge . . . . .	174
10.3.2	RQ2: Common Knowledge and Misconceptions . . . . .	177
10.3.3	RQ3: The Process of Learning . . . . .	182
10.4	Discussion . . . . .	182
<b>11</b>	<b>CS2: Knowledge Structures of Beginning CS Students</b>	<b>187</b>
11.1	Description of the Setting . . . . .	187
11.1.1	Description of the Subject "Informatics" . . . . .	188
11.2	Data Collection & Research Questions . . . . .	190
11.3	Analysis and Results . . . . .	193
11.3.1	RQ1: Prior Knowledge of Beginning CS Students . . . . .	193
11.3.2	RQ2: Effect of CS Education in Secondary Schools . . . . .	198
11.4	Discussion . . . . .	199

<b>12 CS3: Conceptual Knowledge and Abilities</b>	<b>205</b>
12.1 Description of the Setting . . . . .	205
12.2 Data Collection & Research Questions . . . . .	206
12.3 Analysis and Results . . . . .	209
12.3.1 RQ1: Development of Structural Knowledge . . . . .	210
12.3.2 RQ2: Connections Between Knowledge and Abilities . . . . .	215
12.4 Discussion . . . . .	220
<b>V Conclusion</b>	<b>225</b>
<b>13 Summary</b>	<b>227</b>
<b>14 Discussion</b>	<b>231</b>
14.1 Case Studies . . . . .	235
<b>15 Further research</b>	<b>239</b>
15.1 Identifying Threshold Concepts of CS . . . . .	240
15.2 Combining Knowledge Space Theory and Concept Maps . . . . .	241
<b>16 Acknowledgements</b>	<b>243</b>
<b>Appendix A Example Maps for Pathfinder Analysis</b>	<b>245</b>
<b>Appendix B Additional Diagrams for CS1</b>	<b>247</b>
<b>Appendix C Additional Diagrams for CS3</b>	<b>249</b>
<b>List of Figures</b>	<b>259</b>
<b>List of Tables</b>	<b>264</b>
<b>References</b>	<b>265</b>



# Part I

---

## Introduction



# 1 | Problem Setting

This thesis presents data driven methods and the results of empirical research dealing with learning in the context of computer science (CS). As such, it is interdisciplinary in nature, depending both on computer science and on research about education and learning. The focus lies on investigating and visualizing the state and development of structural knowledge. Research concerned specifically with computer science education is a relatively recent field of interest (cf. Fincher & Petre 2004, p. 1), especially when compared to the corresponding research in other subjects like mathematics, physics, history, or languages, for example. These all have a much longer tradition in the curricula of schools and in universities. Consequently the knowledge about and experience in teaching these subjects is far more elaborate and widespread than for computer science, where the subject itself has emerged merely half a century ago. This is especially true for learning how to program:

“Huge numbers of papers have been published in computing education conferences and journals in the past 40 years, so we would expect much to have been learnt about teaching and learning in computing. However, after decades of research, we still have only a vague understanding of why it is so difficult for many students to learn programming, the basis of the discipline, and consequently of how it should be taught.” (Malmi, Sheard, Simon, Bednarik, Helminen, Korhonen, Myller, Sorva & Taherkhani 2010, p. 9)

The advancement of educational research in other subjects is a benefit, of course, as the researchers of computer science education are not forced to repeat every erroneous way of others done in the past. Also, the research about education and learning in general has advanced greatly over the last decades. Constructivism as a learning theory that is backed by findings of Psychology, as presented in more detail in the third chapter, forms the basis of the current understanding of how learning progresses. Therefore, teaching should best adapt accordingly. Luckily, many parts of computer science are well suited for constructivist approaches anyway - as described in (Mühling, Hubwieser & Brinda 2010, p. 60). Concerning the research about education in general, the last decades have brought about a shift in perspective:

“Since the end of the 1980s, the introduction of new oversight strategies for governmental intervention worldwide has led to a

stronger focus on ‘outputs’ and ‘outcomes’ at all levels of the educational system, from elementary through secondary and tertiary education up to vocational and adult education. These outcomes - or the value added to them - are used as criteria for the productivity of entire educational systems, the quality of individual educational institutions, and the learning achievements of individuals. The role of educational research, then, is to render this educational productivity measurable, to develop models that can explain how educational processes take place, evaluate their effectiveness and efficiency, and propose and analyze strategies for intervention.” (Klieme, Hartig & Rauch 2008, p. 3)

Placing the focus more on outcomes also necessitates new ways of testing, which is typically done in the form of competence based assessments in these scenarios. One example of a major testing effort is the international large scale study PISA (PISA 2009 Technical Report 2012, for 2009). Testing competences, as briefly described in chapter 3, requires established models that define what a competence entails and test items that are actually assessing this particular competence. Establishing such models and tests for CS clearly is a valuable and necessary area of research (cf. Linck, Ohrndorf, Schubert, Stechert, Magenheimer, Nelles, Neugebauer & Schaper 2013, p. 1), as is the investigation of teaching competencies specifically for CS (cf. Hubwieser, Magenheimer, Mühlhling & Ruf 2013, p. 1). However, beyond making the educational productivity measurable, the development of explaining models for learning computer science and the analysis of strategies for intervention are also valid research goals, especially when taking into account the current rather barren research landscape concerning CS education. This calls for fundamental research about the inner processes involved in learning and, ultimately, also teaching computer science.

A detailed analysis of theories of instruction reveals, that there are typically three components (cf. Glaser & Bassok 1989, p. 631):

1. A description of the desired goal of the learner concerning competent performances that encompass both knowledge and skills.
2. An analysis of the initial state of these performances, i.e. of the learner’s knowledge and ability.
3. A model of learning that explains the transition from the initial state to the desired state and can be implemented in an instructional setting.



All three components encompass an element of knowledge. Therefore, when analyzing processes of learning and teaching CS, one major aspect is the state and development of (conceptual) knowledge, as defined in chapter 3. To become a competent computer scientist as well as a competent programmer, a person must acquire a certain set of skills as well as a certain body of knowledge. There is no way to become an expert without a rich and highly connected conceptual and factual knowledge. “Thus, one goal of instruction should be to help students acquire expert-like knowledge structures in their domain of study” (Trumpower & Goldsmith 2004, p. 427). Even basic programming skills require factual knowledge about syntax elements and conceptual knowledge about program flow, for instance. Modern constructivist teaching implies that such a body of structured knowledge cannot passively be transported from teachers to learners. Instead, teaching in a constructivist setting requires getting students to actively engage in the process of knowledge construction. Educational research, in turn, must find ways of investigating the idiosyncratically developing knowledge, which is not easily done:

“It should be recognized that we are trying to probe into a person’s cognitive structure and ascertain what concepts and propositions that person has relevant to a particular topic and how are these integrated and organized? This is a profoundly challenging task, and yet it is fundamental to improving teaching and learning in any field. It is also essential for the capture and archiving of expert knowledge [...]” (Cañas & Novak 2006, p. 497)

Investigating conceptual knowledge will remain relevant as Goldstone & Kersten (2003, p. 601) note: “Concepts are useful when they provide informative or diagnostic ways of structuring this world. An excellent way of understanding the mental world of an individual, group or scientific community, or culture is to find out how they organize their world into concepts.”

All of this sets the stage for modern computer science education research and defines the general context of this work. Fincher & Petre (2004, p. 3ff.) identify ten general areas of motivation for further research in computer science education. This thesis is centered in the first of these ten, which concerns the understanding of students:

“The area of student understanding is characterized by investigation of students’ mental and conceptual models, their perceptions and misconceptions. The kinds of question that researchers find motivating in this area are concerned with *why* students have

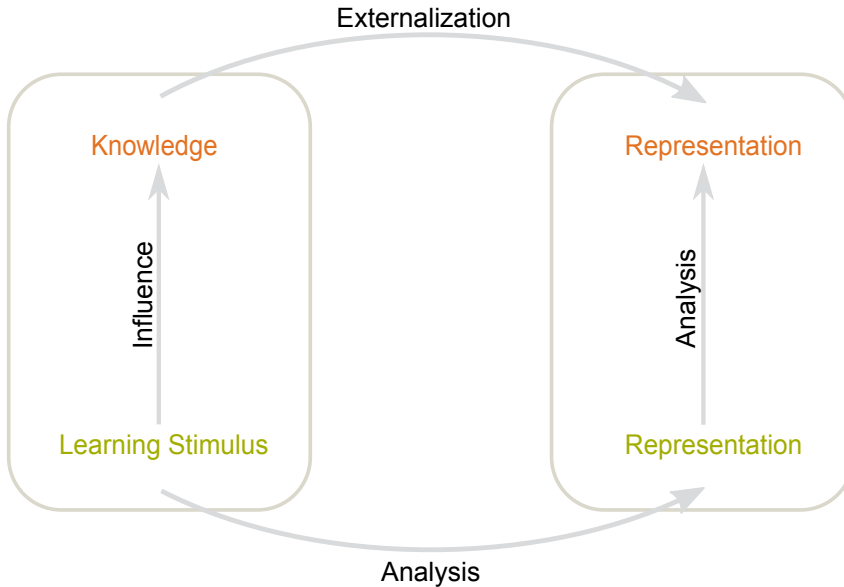


Fig. 1: A general model for monitoring the effects of a learning stimulus on the knowledge of a person. Even though one is generally interested in the development in the left half of the diagram (i.e. the real person), only the right half is accessible to research. The transfers in form of an externalization and the analysis of the stimulus are in general not loss-less and are subject to many influences.

trouble with some of the things they have trouble with, what *distinguishes* good students from bad students, and what the differences are between how students understand things and how experts understand things.”

Since established models of learning are not yet present for CS, a reasonable approach of research is to conduct exploratory studies. It was with this goal in mind, that several investigations were conducted by the research group “Didaktik der Informatik” at the TU München in the years of 2010 to 2013. The investigations all aimed to find out how the conceptual knowledge of learners is influenced by differing forms of computer science education. These forms ranged from an introductory course on programming with only minimal input given, over a lecture, to the investigation in how far attending a (newly introduced) compulsory school subject influences conceptual knowledge about basic CS concepts. While the studies all differed in their respective details, they all had a basic research design in common that is shown in Fig. 1.

The focus of interest always is the current state or development of knowledge at or over some point(s) in time. This knowledge is influenced by educational processes, learning materials, activities of the learner and more - all subsumed under the general notion of "learning stimulus" in Fig. 1. The word "stimulus" was chosen over the word "input" to indicate that, as also explained in the next part, it can not be assumed that the knowledge is "formed" by an input, or that an input is "transported" in some way. The knowledge of a person is not directly observable. It is indirectly observable though by using some form of externalization (see section 3.1.2). This form can be rather direct, like conducting an interview or indirect, by having the person take a multiple choice test, for example. For all of the studies that this thesis is based on, concept maps, as described in chapter 4 and investigated more closely in chapter 6, were chosen as the format of knowledge representation and concept mapping as the process of externalization. The analysis of the stimulus, which may take on such different forms as a textbook, a curriculum, slides used in lectures, a talk given by a teacher, or more is also important as it forms a basis for what can be "looked for" in the knowledge development. Even though there will never be and cannot be a simple "transfer" of knowledge from teacher to student, the divergences between the two nevertheless may point to problem areas of the educational process. There will usually be very many ways to represent the stimulus in a form that is suitable for this kind of analysis; any form will have its advantages and drawbacks. In a similar setting, Trumpower, Sharara & Goldsmith (2010, p. 6) describe a process consisting of three phases: "1) knowledge elicitation, 2) knowledge representation, and 3) knowledge evaluation". Following the descriptions given for each phase, the elicitation corresponds to the externalization of knowledge and the second phase, representation, corresponds to the analysis of the stimulus where they have "suggested careful task analysis or consideration of curriculum documents, textbook content, and other pedagogical material as a starting point"(Trumpower et al. 2010, p. 25). The evaluation is the analysis of the correlation between the two.

While research about CS education has many facets and learning processes are of high importance, also the evaluation of educational processes (i.e. teaching) is pivotal. When using the above schema of investigating knowledge in this context, a shift in perspective must occur: Instead of externalizing and analyzing the knowledge of a single person, the analysis must focus on many persons of the same educational process. It is this combined evidence in which the artifacts of an educational process can show. The same is done in other studies that are concerned with evaluating educational processes like PISA - measurements of many individuals are combined for analysis. A simple example of an exam, e.g. in school shows how results can be interpreted from different points of view with different outcomes: On the one hand, a person can be interested in the particular result of one student. A teacher,

for example, might be interested to know that this person failed the exam, or failed multiple exams in a row. On the other hand, a person can also be interested in the the results over all students. The school's principal might be interested to know that almost all students failed the exam. In the first case, no particular insight about the exam can be gained - unless the student would usually not fail an exam. In the second case, no particular insight about the individual students can be gained, in general.

This work presents - in chapter 7 - a new way of analyzing the externalized knowledge (with concept maps) in the light of measurements of many persons and also of several points in time. It is the result of three research studies that are presented in chapters 10 to 12. The ideas developed over the course of these studies and are summarized in this work. They are inspired by the ideas of data mining.

In recent years, *Educational Data Mining (EDM)* has emerged as a new discipline in the field of educational research. It is “[...] concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in. [...] Educational data mining methods are drawn from a variety of literatures, including data mining and machine learning, psychometrics and other areas of statistics, information visualization, and computational modeling” (Baker & Yacef 2009, p. 4). According to Baker (2010, Table 1), the five primary categories of Educational Data Mining and their respective goals are:

**Prediction** Develop a model which can infer a single aspect of the data (predicted variable) from some combination of other aspects of the data (predictor variables).

**Clustering** Find data points that naturally group together, splitting the full data set into a set of categories.

**Relationship Mining** Discover relationships between variables.

**Discovery with models** A model of a phenomenon developed with prediction, clustering, or knowledge engineering, is used as a component in further prediction or relationship mining.

**Distillation of data for human judgment** Data are distilled to enable a human to quickly identify or classify features of the data.

The research presented in this work is concerned with the second, third and fifth category. The chosen methods of analysis are described in chapter 5, applied to

---

the particular context in chapter 7 and tested on real-world scenarios in chapters 10 to 12. While the externalization of knowledge, its analysis and the analysis of the learning stimuli don't require computer support per se, the use of data mining techniques and computer supported analysis allows the processing of large bodies of data. Since such large amounts of data cannot be processed manually in reasonable time, computer support is more than a way of effectively saving time: It allows new insights into the (often statistical) properties of the data. Using larger amounts of data often results in new findings that cannot be found in small samples due to noise or statistical insignificance. The proposed methods can all readily be implemented in software; for the most part, using a computer (e.g. for drawing concept maps) is a natural extension of the task and allows enlarging the amount of data that can be handled greatly. In this way, it is easily attainable to collect and immediately analyze data from many students worldwide without additional personal labor costs. Therefore, all proposed methods have been implemented as part of this thesis, as presented in chapter 8.



## 2 | Detailed Overview

The research questions underlying this thesis are twofold. First, there is a general question that drove the development of the method presented in this thesis:

*How can the knowledge structures of a group of persons be investigated with regard to common elements and differences between the individuals' structures?*

Derived from this general question while also taking into account the historical development of the thesis described below, the following more specific research questions are answered in the third part:

- 1. How can methods of data mining be applied to sets of concept maps in order to identify common elements and differences between the individual maps?*
- 2. How can software support the workflow of the research design presented in Fig. 1?*

In addition to this question, the different research projects all had their own, specific research questions dealing with computer science education. These are presented in short below and in greater detail in the fourth part separately for each case study, as they are dependent on the particular context of each study.

The rest of this thesis is divided into three major parts. First, the literature of related prior research is presented. It forms the theoretical background for the method presented in chapter 7. Instead of simply focusing on the relevant details, the chapters of the second part are trying to give a slightly more complete overview of the relevant aspects in order to present everything that is necessary to understand the rest of the thesis without having to resort to the references too often. It encompasses a rather broad field, as described in more detail below, and ranges from psychological and neurological models of learning over learning theories to statistical methods of analyzing pattern in data. The next part then contains the first half of the contributions of this work: An investigation of concept maps and the development of the notion of concept landscapes. Analysis methods that are suitable for working with them are presented as well as the software “tool-chain” that has been developed in the course of this thesis. These contributions are described in theory first and are then applied to actual research studies, which form the second half of the contributions of this thesis; the three case studies are based on actual investigations. Each study is presented self-contained in one chapter. The structure of each chapter is identical and resembles the organization of a research paper. In

each of the studies, concept maps were collected from students in order to analyze conceptual knowledge. In the course of these investigations, the focus shifted more and more from the investigation of personal knowledge structures towards the common elements of groups of persons. This development then culminated in the idea of concept landscapes that formalize this approach. The following gives a brief summary of the contexts in the temporal order that they were conducted:

**Knowledge development in a setting with minimal input** One of several preparatory courses offered to beginning CS students at the TU München deals with basics of object-oriented programming (OOP), as presented in (Hubwieser & Berges 2011). The course design offers only minimal input and instead focuses on self-guided, active application of programming concepts. The development of the knowledge in the light of this setting and the interplay between knowledge and programming abilities were the central aspects of this study, done in 2010. The original study has been published in (Berges, Mühling & Hubwieser 2012). The results show, that the minimal input together with practical exercises are enough to initiate visible alterations of students' mental models. Also, cluster analysis identifies groups of students with markedly different developments in their mental models. Also, the interrelation between conceptual knowledge and programming abilities is seemingly not as straight-forward as one would hope for.

**Investigation of knowledge development in a lecture** An introductory lecture in computer science for non-majors was investigated in 2010. Of particular interest was how conceptual knowledge structures of core concepts of object orientation (OO) develop over time. The results of the original study have been published in (Hubwieser & Mühling 2011c). The study shows, that the learning of the students visibly progresses over time. Nevertheless, the process is rather complex and misconceptions also remain prevalent throughout the lecture.

**Influence of compulsory CS education in secondary schools on beginning students** Bavaria, one of the federal states of Germany, introduced a new subject "Informatics", as described e.g. in (Hubwieser 2012). As part of a larger research project in 2011, the conceptual knowledge of beginning CS students regarding central concepts of the curriculum of this subject have been analyzed. Especially, the differences to students who did not attend this compulsory school subject were of interest. The results have not been published previously. The influence on the school-subject on the mental models is clearly visible. The relevant prior-knowledge



of the beginning students is surprisingly complex. Also, there are groups identifiable with regard to the structure of the knowledge that exist regardless of the influence of the school subject.

The idea of concept landscapes has been developed over the course of these research projects. It was therefore not fully present (or named) at the time of the studies. The analysis methods that were used during the studies were revisited and adapted, afterwards, in order to make them compatible with the new approach of concept landscapes. Then, the data of each case study was re-analyzed in the course of writing this thesis in the light of the novel techniques. So, the original projects are serving as case studies for the practical applications of the idea in this thesis, but historically, they came first. Several hundred students have taken part in the studies and the results are of actual value for computer science education. Therefore, they are a fundamental part of this thesis and significantly more than just “toy examples”.

Since the idea behind concept landscapes is based on data mining, computer support presents an important aspect for each of the three subtasks identified in Fig. 1: The externalization in form of concept mapping as well as the analysis of both the input and the externalized knowledge can all be supported effectively with software. The tool-chain that has been developed for this contains the following three projects:

**CoMapEd** An online editor that allows creation of concept maps in a browser and researchers to easily and comfortably collect numerous concept maps in the course of a survey.

**ConEx** A *Java*<sup>1</sup> based program that analyzes text data and extracts salient concepts as well as sentences in which these concepts appear. It can be used to automatically analyze large amounts of textual input for educational processes.

**CoMaTo** A package for the statistical language *R*<sup>2</sup> that allows performing the analysis methods presented in this thesis.

Due to the historical development described above, the concept mapping editor was not used for the actual studies. However, CoMapEd has been used in other contexts. The data of this is analyzed and presented in chapter 6. CoMaTo has been used for the analysis of the case studies and is, partly, a collection of the

---

<sup>1</sup><http://www.java.com>

<sup>2</sup><http://www.r-project.org>

software used originally to analyze the studies. ConEx also has been developed too late to have been of actual use in the studies, but a comparison between the results obtained manually in the case study and ConEx is presented.

The work in this thesis draws upon different research areas. Fig. 2 illustrates the interconnections to the research of this thesis. First, there is computer science, where methods of analysis, most notably data mining are derived from. Also, the software developed in the course of this thesis is, in the end, based on computer science. Finally, it also provides the context for the case studies. Second, there is educational research which provides much of the theoretical background for the research questions this thesis poses. Also methods of investigating knowledge are taken from this area. Finally, third, there is didactics of informatics, which sets the context for the research questions of this thesis and provides theoretical background for the results. These results are then again relevant twofold: The results of the analysis in the case studies are relevant first and foremost for didactics of informatics while the methods presented are also relevant for educational research in general.

This interplay of different areas is typical for research of the educational aspects of a specific subject. Shulman (1986, p. 10) identifies three types of knowledge relevant for teachers, namely “(a) subject matter knowledge, (b) pedagogical content knowledge, and (c) curricular knowledge”. While the latter is mostly covered by educational research, it is pedagogical content knowledge which is mostly the result of research in didactics of a specific subject matter (cf. Hubwieser et al. 2013, p. 1).

The terms *computer science* and *informatics* are used interchangeably throughout this thesis. In German, “Informatik” is often translated as computer science, whereas the term informatics often is seen as comprising more than just computer science. However, as this thesis is not concerned with the details of the different subjects, informatics and computer science are both taken to mean the German term Informatik, which is probably closer to the term informatics than to the term computer science, even though the latter is more widely used in English literature.

Also, for all software that has been used in the course of this work, a link to the relevant website is given as a footnote at the first appearance in the text.

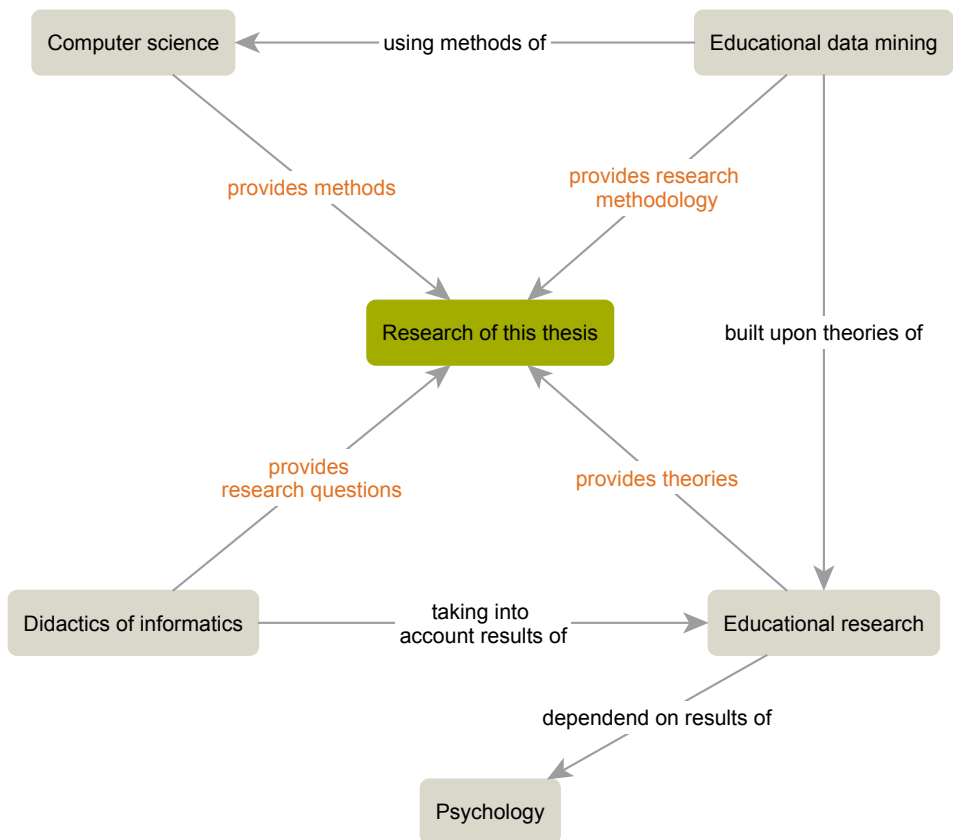


Fig. 2: The areas of research that provide the basis for this thesis.



## **Part II**

---

# **Theoretical Background and Related Work**



## 3 | Knowledge and Learning

“All men by nature desire to know.’ With these words, Aristotle began his *Metaphysics*. But what is knowledge? What do people have inside their heads when they know something? Is knowledge expressed in words? If so, how could one know things that are easier to do than to say, like tying a shoestring or hitting a baseball? If knowledge is not expressed in words, how can it be transmitted in language? How is knowledge related to the world? And what are the relationships between the external world, knowledge in the head, and the language used to express knowledge about the world?” (Sowa 1984, p. 1)

Since this thesis deals with the investigation of knowledge structures, it is inevitable to take a look at the concept of “knowledge”. Dating back to Aristotle and beyond, there have been countless attempts, theories, and results regarding the investigation and classification of knowledge. While some fields of research are treating knowledge as an abstract entity on its own, others are concerned with memory rather than with knowledge. In one way or another, the two are equivalent, since knowledge and the reasoning about and inquiry into knowledge are always tied to the human brain as the creator, processor, distributor, and analyzer of knowledge.

The next three sections present the theoretical background and related work concerning the organization of knowledge, the acquisition of knowledge (learning) and the assessment of knowledge as far as it is relevant for this work. Psychological, neurological, and pedagogical results are taken into account.

### 3.1 Knowledge

This section presents relevant literature concerning the organization of knowledge. This encompasses both the organization of knowledge in the brain as well as the organization of knowledge outside of the human mind as far it is relevant for this work.

### 3.1.1 Psychological Foundations

Knowledge organization from a psychological or neurological point of view encompasses results and theories regarding the human brain and the way that knowledge is stored in and retrieved from it. Historically, theories about memory organization are often based on observing human behavior and subsequently establishing a model of (parts of) the mind that explains the observed phenomena. Alternatively, from a neurological point of view it is also possible to observe not human behavior but actual activity of the brain as an organ. For example, the structure of the brain into several lobes or regions (cf. Sousa 2009, p. 16) and measuring activity of these regions during different tasks provides a theory about the structure of the mind as an organ. In contrast, the “information processing model” (cf. Sousa 2009, p. 38ff.) is a more abstract separation of the human mind into functional components (based on observation of humans) which may or may not map to actual regions of the brain. Following this popular model, the processing of information is based on the *immediate memory*, which is concerned with sensory input from the outside world (cf. Sousa 2009, p. 42) and the *working memory* which is where the conscious act of thinking occurs (cf. Sousa 2009, p. 45). Both of these parts form the *short-term memory*, while storage and retrieval of information is the task of *long-term memory* (cf. Sousa 2009, p. 51).

Since this section is concerned with knowledge organization, it focuses on details of the long-term memory. Concerning the organization of this memory (or storage system), a general distinction can be made between *declarative memory* and *non-declarative memory* (cf. Novak 2002, p. 553). “Declarative memory (also called *conscious* or *explicit* memory) describes the remembering of names, facts, music, and objects [...], and is processed by the hippocampus and cerebrum. [...] Declarative memory can be further divided into *episodic memory* and *semantic memory*” (Sousa 2009, p. 81). Whereas episodic memory mostly deals with events of our lifetime and has a strong connection to the passing of time, it is the semantic memory that holds what is commonly called “declarative knowledge” in the context of learning and teaching, like facts or knowledge about concepts and their interrelations. Squire (1987, p. 152) describes the specifics of declarative memory even more succinctly: “Declarative memory is memory that is directly accessible to conscious recollection. It can be declared. It deals with the facts and data that are acquired through learning”. The separation of these types of memory in the brain has been shown in medical studies (cf. Goldstein & Vanhorn 2011, p. 158f.), however, there are also interrelations between episodic and semantic memory. For example Goldstein & Vanhorn (2011, p. 160) note that “semantic memories that have personal significance are easier to remember than semantic memories that



are not personally significant". In addition to this "semantic memory can influence how people allocate their attention".

Non-declarative memory also encompasses several sub-categories, most notably *procedural memory*. "Procedural memory refers to the learning of motor and cognitive skills and remembering *how* to do something [...]" (Sousa 2009, p. 82). There are other classifications of the components of human memory as well, which are typically refinements of the general distinction made above. For example, Trumpower & Goldsmith (2004, p. 430) distinguish between *propositional* and *configural knowledge*, the former referring to facts and the latter meaning the interconnected structure of knowledge. In an attempt to summarize different studies that focus on the task of problem solving, de Jong & Ferguson-Hessler (1996) offer an attempt for a classification system concerning types of knowledge. It distinguishes between the *type* and *quality* of knowledge, which according to the authors form two separate dimensions. The types that are distinguished are *situational*-, *conceptual*-, *procedural*- and *strategic knowledge*. The qualities are *level*, *structure*, *automation*, *modality* and *generality* (de Jong & Ferguson-Hessler 1996, Table 1).

Conceptual knowledge, again, is defined as "static knowledge about facts, concepts and principles that apply within a certain domain" (de Jong & Ferguson-Hessler 1996, p. 107). Situational knowledge and strategic knowledge are concerned with "situations as they typically appear in a particular domain" (de Jong & Ferguson-Hessler 1996, p. 106) and "students organizing their problem-solving process" (de Jong & Ferguson-Hessler 1996, p. 107) respectively. The qualities of knowledge each represent a certain attribute that can take on different values (cf. de Jong & Ferguson-Hessler 1996, Table 1):

**Level** Is it *surface* or *deep* knowledge?

**Structure** Does the knowledge consist of *isolated elements* or is it *structured knowledge*?

**Automation** Is the knowledge *declarative* or *compiled*?

**Modality** Is the knowledge stored in *verbal* or *pictorial* form?

**Generality** Is the knowledge *general* or *domain specific*?

While surface level knowledge is "stored in memory more or less as a copy of external information" (de Jong & Ferguson-Hessler 1996, p. 107), deep level knowledge is "firmly anchored in a person's knowledge base and [...] has been translated to basic concepts, principles, or procedures [...] [and] is different from

the concrete appearance of the external information from which it stems” (de Jong & Ferguson-Hessler 1996, p. 107). The quality of structure distinguishes between elements of knowledge that are “loosely connected [...] or structured in a logical way” (de Jong & Ferguson-Hessler 1996, p. 108) and is related to the level of knowledge: “Only the introduction of deep elements makes possible the generalizations and abstractions that are required for the [...] building of a hierarchical structure” (de Jong & Ferguson-Hessler 1996, p. 108). Declarative knowledge is defined as above, in contrast to *compiled knowledge* that “is tacit, implicit, or not easily expressed” (de Jong & Ferguson-Hessler 1996, p. 109). Modality refers to the representation of knowledge “as a set of either propositions or images” (de Jong & Ferguson-Hessler 1996, p. 109). Generality refers to the fact that knowledge or strategies (e.g. heuristics) “may be general and domain independent [...] but frequently they are bound to a domain”(de Jong & Ferguson-Hessler 1996, p. 109).

The structure of organized knowledge is important for the development of learners and should therefore receive attention. “In general, structured knowledge enables inference capabilities, assists in the elaboration of new information, and enhances retrieval. It provides potential links between stored knowledge and incoming information, which facilitate learning and problem solving” (Glaser & Bassok 1989, p. 648). Trumpower et al. (2010, p. 5) also state that “experts possess more knowledge and, perhaps more importantly, better organize knowledge than novices”. The term *knowledge structure* or *cognitive structure* (cf. Ausubel 1968, p. 10) will be used to denote the structural organization of a person’s conceptual knowledge.

The very name “conceptual knowledge” and its definitions rely on the notion of a *concept* which is also fundamental for the later definition of concept maps. From a psychological perspective, Solomon, Medin & Lynch (1999, p. 99) state that “[c]oncepts are the building blocks of thought. How concepts are formed, used, and updated are therefore central questions in cognitive science”. The following definition will be used: “A concept is a mental representation that is used for a variety of cognitive functions, including memory, reasoning, and using and understanding language.” Goldstein & Vanhorn (2011, p. 240)

The mental representation can be of something that exists as a concrete object, like a tree (cf. Solomon et al. 1999, p. 101) or an abstract notion, like a disease (cf. Solomon et al. 1999, p. 100) or a color (cf. Goldstein & Vanhorn 2011, p. 245). Mental representations are usually connected to a mental function or process that the representation is useful for (cf. Goldstone & Kersten 2003, p. 603). For concepts, this process is often *categorization*: the classification of entities into *categories*. Categories and concepts are closely related. Categorization happens through concepts (cf. Solomon et al. 1999, p. 99), in other words for each category

there must also be a concept. The category then refers to the classified set of entities in the category whereas the concept refers to the abstract notion of what this category entails. "Categorization is not an end in itself, but rather it serves to connect old to new: categorizing novel entities allows the cognitive system to bring relevant previous knowledge to bear in the service of understanding the novel entity. [...] Not only are new entities understood in terms of old, but new entities also modify and update concepts. That is concepts support learning" (Solomon et al. 1999, p. 99). Especially the first part is important for the theories of conceptual change that are presented below. Categories are organized in hierarchies (cf. Goldstein & Vanhorn 2011, p. 247). There are other functions beyond categorization as well that are based on concepts, like *inference*, *reasoning* (cf. Solomon et al. 1999, p. 99) and *integration* which "refers to the process of finding a relationship that meaningfully links two concepts together" (Solomon et al. 1999, p. 102). The process of integration, in turn "appears central to inductive reasoning" (Solomon et al. 1999, p. 103)

In general, the organization of concepts in memory is assumed to be pivotal for the quality of a person's knowledge (cf. Trumpower et al. 2010, p. 5). Without a structured connection to others, a concept will, in general, not be kept in long term memory (cf. Sousa 2009, p. 88). As Ruiz-Primo & Shavelson (1996, p. 570) point out:

"Most cognitive theories share the assumption that concept interrelatedness is an essential property of knowledge. [...] As expertise in a domain is attained through learning, training and/or experience, the elements of knowledge become increasingly interconnected. [...] Assuming that knowledge within a content domain is organized around central concepts, to be knowledgeable in the domain thus includes having a highly integrated structure among these concepts."

An expansion on the view of knowledge organization is *schema theory*. "A schema is a high-level conceptual structure or framework that organizes prior experience and helps us to interpret new situations. The key function of a schema is to provide a summary of our past experiences by abstracting out their important and stable components [...] a distinguishing feature of schemas is that they are structured mental representations made up of multiple components" (Gureckis & Goldstone 2011, p. 725). Schema theory holds that the mind possesses schemas for various situations that allow a very fast information processing, as the schema is retrieved from memory as one chunk.

### 3.1.2 Epistemological Foundations

While the last section dealt with the organization of knowledge inside the brain, this section deals with how knowledge can be documented outside the human mind. This capacity is essential for teaching (cf. Cooke 1994, p. 801), or, more generally, preserving the knowledge of each human generation for the next. What sets the epistemological perspective on knowledge apart from the psychological view is that the documentation of knowledge can also arise from the mental models of *many* persons (cf. Cooke 1994, p. 821) instead of just the mental model of one person. This aspect is central for this thesis. Also, as Cooke (1994, p. 802) notes, knowledge can be *elicited* from other sources than the human mind, e.g. texts - which is done in this thesis as well. "*Knowledge elicitation* is the process of collecting from a human source of knowledge, information that is thought to be relevant to that knowledge" (Cooke 1994, p. 802). It is often part of a process called *knowledge acquisition*. "The overall goal of knowledge acquisition is to externalize knowledge in a form that can be implemented in a computer" (Cooke 1994, p. 802). Often, it is specifically meant in the context of expert systems and expert knowledge, though (cf. Shaw & Woodward 1990, p. 179). The term *externalization* of knowledge sometimes refers to only making tacit knowledge explicit (cf. Dierkes 2001, p. 495), however, in this work, *externalization* will refer to a general process of elicitation of any form of knowledge - usually semantic, declarative knowledge - from humans regardless of their expertise.

Following the constructivist view presented below, it is paramount to acknowledge that: "[K]nowledge acquisition is modeling or construction, not mining. Therefore, the result of knowledge acquisition is a model. Like all models it represents the object of the modeling enterprise to different degrees of accuracy" (Cooke 1994, p. 802). Or put differently, the object of interest - the mental model - is by itself unobservable. Instead, only a model of this model can be observed (cf. Shaw & Woodward 1990, p. 184). Each method of elicitation and each method of representation is making assumptions about the knowledge itself (cf. Shaw & Woodward 1990, p. 180). The process of externalization is also subject to uncertainties. The influences that incur these uncertainties are numerous, like problems in communication or the amount to which the knowledge that is to be externalized is compiled and more (cf. Cooke 1994, p. 803); also personal variables, like the degree of introversion have been found to influence the process (cf. Hoffman, Shadbolt, Burton & Klein 1995, p. 146). While all methods inherently suffer from these uncertainties, the specific influences and their extent are dependent on the method and context. The same goes for the requirements on the elicitor (e.g. to be a subject-matter expert) or the analysis methods that are suitable afterwards (cf. Cooke 1994, p. 814).

There are many different methods of elicitation and ways of representation. Mandl & Fischer (2000) present an overview over several graphical representations. Hoffman et al. (1995) present a classification of different methods. Cooke (1994, p. 805ff.) also presents a classification that identifies three different families of elicitation techniques: "Observations and interviews", "Process tracing", and "Conceptual techniques".

A technique of the last family "produces representations of domain concepts and their structure or interrelations" (Cooke 1994, p. 821) and is especially suited for the aggregation of knowledge from many persons (cf. Cooke 1994, p. 821) and automation (Cooke 1994, p. 835), making them especially relevant for this thesis. The technique of concept mapping, described in the next chapter, is a representative of the type "graph construction" that belongs to the group of "structural analysis" techniques in this family (cf. Cooke 1994, Table 3). It has been chosen as the method of externalization in this thesis. The reasons for this are given in chapter 4 and chapter 6.

While the "differential access hypothesis" assumes that different methods of elicitation tap into different types of knowledge (cf. Hoffman et al. 1995, p. 142), the approach of externalizing the structure of conceptual knowledge is considered valuable beyond that specific type of knowledge: "Collectively, these studies suggest that [...] [assessing structural information of conceptual knowledge] allows valid inferences to be made about overall domain knowledge across a diverse array of domains, ranging from those that are more procedural (e.g., computer programming) to those that are more conceptual (e.g., evolution)" (Trumpower et al. 2010, p. 10).

The chosen representation also affects what types of knowledge can be documented. Giving an overview of all kinds of knowledge representations and their respective strengths and weaknesses is beyond the scope of this work, however. The chosen method of concept mapping determines the representation (concept maps) anyway. For the analysis methods, different representations of the same information are also used, based on the requirements of the methods.

## 3.2 Learning

"Obviously, all knowledge constructed in a discipline is first constructed in some individual's cognitive structure. To understand how knowledge is constructed in any field, it is therefore essential to understand how individual human beings construct knowledge. To understand how an individual constructs his or her conceptual

frameworks, we need to understand the psychology underlying human meaning making.” (Novak 2002, p. 562)

Beyond the organization of knowledge inside the mind, there are also the elementary processes of creating and altering this knowledge. Since humans are born with almost no knowledge whatsoever, everything must be developed through experience and learning. In regards to the process of learning, it is again possible to either focus on the mind or brain itself, or to focus on the person, possibly also in a larger context of not isolating a single person but acknowledging the society and environment in which the person lives while learning. While the psychological foundations are primarily concerned with the former, the pedagogical learning theories of learning are usually concerned with the latter. There are numerous learning theories, especially when also considering their historical developments. All theories usually are more or less heavily influenced by the zeitgeist of their date of creation. This section focuses on the basic psychological processes of learning and on currently established learning theories that are supported by these.

### 3.2.1 Psychological Foundations

Learning, especially concerning declarative knowledge, encompasses both the storing of new information and the reorganization or alteration of information already present in the memory. Following the distinction in memory organization, “the acquisition of new declarative knowledge, development of a cognitive skill, organization of knowledge into more effective representations, and discovery and inference of new information are differentiated forms of learning, and their characterization varies” (Glaser & Bassok 1989, p. 634).

The modification of knowledge, as also postulated by current learning theories, is fundamentally personal: “Learning is highly idiosyncratic and progresses over time” (Cañas & Novak 2006, p. 495). Knowledge is actively constructed by each person’s memory subjectively. Nevertheless, the structure of the knowledge that is created, with increasing expertise and, depending on the subject domain, often shows a similar configuration across many persons (cf. Trumpower & Goldsmith 2004, p. 426f.). Even more, this configuration is typically similar to those of experts in that domain (cf. Ausubel 2000, p. 76). Also Trumpower & Goldsmith (2004, p. 441) note that the structural organization of knowledge may have an influence on the ability of beginners to learn new material of a subject domain.

The addition of new information into long-term memory - a process that is often called *retention* (cf. Sousa 2009, p. 86) - is influenced by many factors outside of

the actual memory (e.g. motivation). Concerning the actual processes in memory, the retention of conceptual knowledge is critically dependent on the *rehearsing* of new information (cf. Sousa 2009, p. 86). There are two types of rehearsal: *Rote rehearsal* which “is used when the learner needs to remember and store information exactly as it is entered into working memory” (Sousa 2009, p. 87) and *elaborative rehearsal* for “when it is more important to associate the new learnings with prior learnings to detect relationships” (Sousa 2009, p. 87), see also (Goldstein & Vanhorn 2011, p. 174f.). This distinction is also being made by the theory of meaningful learning, presented below. Learning in the context of educational research is often seen as elaborative learning. In this case the prior knowledge becomes important. “The prevailing view of cognitivists today is that humans store knowledge as associative networks of ideas, concepts, procedures, and other forms of knowledge. During learning, new knowledge is integrated into the network by linking it to semantically relevant prior knowledge” (Trumpower et al. 2010, p. 5).

With increasing training and expertise in a given field, the conceptual knowledge will get restructured and is then accessible in bigger chunks of connected knowledge. Also, as skills develop, the necessary knowledge will get compiled into procedures. As Glaser & Bassok (1989, p. 634) put it: “At various stages of learning, there exist different integrations of knowledge, different degrees of proceduralized and compiled skill, and differences in rapid access to memory [...]. These differences signal advancing expertise or possible blockages in the course of learning”.

### 3.2.2 Constructivism

Constructivism as a learning theory is best understood as opposite to the theories of Behaviorism and Cybernetics, where the prevalent idea is that teaching is nothing more than a transfer from the outside world into the mind of the learner. As Glasersfeld (1983, p. 41) puts it:

“Ten or 15 years ago, it would have been all but inconceivable to subject educators or educational researchers to a talk that purported to deal with a theory of knowledge. Educators were concerned with getting knowledge into the heads of their students, and educational researchers were concerned with finding better ways of doing it. There was, then, little if any uncertainty as to what the knowledge was that students should acquire, and there was no doubt at all that, in one way or another, knowledge could be transferred from a teacher to a student. The only question

was, which might be the best way to implement that transfer – and educational researchers, with their criterion-referenced tests and their sophisticated statistical methods, were going to provide the definitive answer.”

As a contrast, the central aspect of Constructivism is that “[k]nowledge [...] cannot be imposed or transferred intact from the mind of one knower to the mind of another. Therefore, learning and teaching cannot be synonymous: we can teach, even well, without having students learn.” (Karagiorgi & Symeou 2005, p. 18)

Fundamentally, Constructivism is built upon two principles (Glaserfeld 1989, p. 162):

1. “knowledge is not passively received but actively built up by the cognizing subject” and
2. “the function of cognition is adaptive and serves the organization of the experiential world, not the discovery of ontological reality”.

The first principle, when followed strictly, does imply that there is no “true” knowledge, since all knowledge is only constructed inside a person’s mind and cannot, in general, be transferred in or out of that mind. Consequently, Glaserfeld (1983, p. 65) describes knowledge in the constructivist sense as: “A knowledge that fits observation. It is knowledge that human reason derives from experience. It does not represent a picture of the ‘real’ world but provides structure and organization to experience. As such it has an all-important function: it enables us to solve experiential problems”. It is, however, especially this second principle, that separates Constructivism from other theories. “The revolutionary aspect of Constructivism lies in the assertion that knowledge cannot and need not be ‘true’ in the sense that it matches ontological reality, it only has to be ‘viable’ in the sense that it fits within the experiential constraints that limit the cognizing organism’s possibilities of acting and thinking” (Glaserfeld 1989, p.162).

Historically, the theory is based on studies by Jean Piaget (Piaget 1929). Meanwhile, research has offered enough insight into the neurological mechanisms of learning, that the constructivist theory can justifiably be seen as an explanatory model of human learning and knowledge construction. Sabitzer (2011) provides insights into the neurological aspects of teaching computer science, stating that learning is the process of subjectively constructing knowledge. Wittrock (1992, p. 536) states from the perspective of cognitive research: “[L]earning is not the internalization of information given to us whole by experience or analyzed by a teacher. Instead,



learning consists of the active generation of meaning, not the passive recording of information.”. Finally, Altenmüller, Gruhn, Parlitz & Liebert (2000, p. 49) present the results of a study in music education. The results show, that the activation patterns of the brain differs between groups that were taught using “verbal explanations, visual aids, notations, verbal rules and some musical examples which were played to the subjects, but never sung or performed [and a second] group who participated in musical experiences for establishing genuine musical representations by singing and playing, improvising with corresponding rhythmic and tonal elements or performing examples from the musical literature”. The second group also showed superior results in an assessment task.

There are many consequences for teaching and educational research when adopting this theory. “Above all, it will shift the emphasis from the student’s ‘correct’ replication of what the teacher does, to the student’s successful organization of his or her own experience” (Glaserfeld 1983, p. 69). Teaching in a constructivist setting also implies that the knowledge structure of individuals may differ from what the physical world dictates or differ from what experts of a field are agreeing upon. In this case, it seems reasonable to assume that the learner will benefit from attempts that result in a modification of the knowledge structure. Constructivism also puts a limit on the results one can expect from assessing a person’s knowledge by comparing it to expert knowledge - which is also in accordance with the description of “deep” knowledge presented in the first section of this chapter. Radical Constructivism places difficult burdens on educational systems. The following two quotes of the same publication are an example of the difficulty of reconciling a constructivist mode of learning with the long established tradition of education at, for example, universities.

“I now believe that the days of straight lecturing in introductory science courses are numbered. We can no longer afford to ignore that inefficiency of the traditional lecture method, regardless of how lucid or inspiring our lectures are. The time has come to offer our students in introductory science classes more than a mere regurgitation of printed material.” (Mazur 1996, p. 14)

“[...] in the sciences, as in the humanities, the first exposure to new material should come from reading printed material.” (Mazur 1996, p. 13)

### 3.2.3 Conceptual Change

“[E]ffective learning changes the way we see the world- The acquisition of information in itself does not bring about such a change, but the way we *structure* that information. Thus, education is about *conceptual change*, not just the acquisition of information.” (Biggs & Tang 2011, p. 23). The theory of conceptual change is typically attributed to Posner, Strike, Hewson & Gertzog (1982) and revised by Strike & Posner (1992). Learning can, very generally, be described as change. A change of a person’s knowledge structure, a change of a person’s behavior or attitudes can all be the results of learning processes. The learning theories presented so far acknowledge this - for example meaningful learning as described in the next section by its very definition is the change of an existing mental model. The idea of conceptual change in contrast is subtly different in that it puts the focus on the very act of change instead of just acknowledging its existence. The change of a person’s knowledge structure can be additive in the sense that an existing model of the world is expanded in the light of new experiences or facts or it can require that the model has to be restructured or even completely abandoned because it is not compatible with new knowledge. Piaget calls these two ways of change *assimilation* and *accomodation* (cf. Geber 2006, p. 5), (cf. Goldstone & Kersten 2003, p. 603). Duit & Treagust (2003, p. 672) note that “[t]he most common analysis is that there are two types of conceptual change, variously called weak knowledge restructuring, assimilation or conceptual capture and strong radical knowledge restructuring, accommodation or conceptual exchange”.

It is especially this last part that plays an important role in conceptual change as, naturally, there is an impediment to a restructuring or abandoning of ones mental models, in general. This becomes especially important in science education, since typically everybody has a preconceived notion about how, for instance, the physical world works. As Duit & Treagust (2003, p. 671) put it: “Findings from many studies over the past three decades show that students do not come into science instruction without any pre-instructional knowledge or beliefs about the phenomena and concepts to be taught. Rather, students already hold deeply rooted conceptions and ideas that are not in harmony with the science views or are even in stark contrast to them”. Such a model that is not in harmony with the science view has been given several names (cf. Novak 2002, p. 555), often *misconception*. Novak (2002, p. 555) suggests “Limited or Inappropriate Propositional Hierarchies (LIPH’s)”.

Just like other learning theories, the theory of conceptual change has direct implications for teaching. “The classical conceptual change approach involved the

teacher making students' alternative frameworks explicit prior to designing a teaching approach consisting of ideas that do not fit the students' existing ideas and thereby promoting dissatisfaction. A new framework is then introduced based on formal science that will explain the anomaly" (Duit & Treagust 2003, p. 673). However, Novak (2002, p. 562) also points out, that albeit the process of instruction can influence conceptual change, it is inherently a decision of the learner: "What becomes central to 'conceptual change' from my perspective is the necessity for meaningful learning to occur. [...] The fundamental challenge to 'conceptual change teaching' is therefore to help learners understand how they must choose to modify their concept and propositional hierarchies [...]. Changing their 'conceptual ecology' requires that the learner recognize explicit ways where their concept/propositional frameworks are limited, inappropriate or poorly organized into hierarchies". Glaser & Bassok (1989, p. 642) note that "conceptual change is self-directed, in the sense that humans are intrinsically motivated to understand the world around them".

What is especially interesting is the event of a major change in a knowledge structure. *Threshold concepts* provide a theory for this. Meyer & Land (2006, p. 3) describe them as: "[A]kin to a portal, opening up a new and previously inaccessible way of thinking about something. It represents a transformed way of understanding, or interpreting, or viewing something without which the learner cannot progress. As a consequence of comprehending a threshold concept there may thus be a transformed internal view of subject matter, subject landscape, or even world view". For example, the concepts of complex numbers and limits are seen as threshold concepts in mathematics (cf. Meyer & Land 2006, p. 5). The characteristics of thresholds concepts within a discipline are as follows (cf. Kinchin, Cabot, Kobus & Woolford 2011, p. 2):

**Transformative** The learning results in a change of perception, values, or attributes.

**Irreversible** The change is unlikely to be forgotten.

**Integrative** New interrelations to other concepts of the subject are discovered.

**Bounded** The concept is pivotal to an area of an academic discipline.

**(Potentially) troublesome** The concept typically provides problems to students when they are presented with the new perspective that it offers.

Threshold concepts therefore offer a new aspect to the theory of conceptual change, insofar as they provide a reasoning for problems in learning that is mostly irrele-

vant to the prior knowledge and is only centered around the characteristics of the threshold concept itself.

Since the focus of conceptual change lies on the change of knowledge structures, the theory must also make assumptions about the organization of knowledge. Özdemir & Clark (2007) present an overview over two opposing theories on conceptual change regarding the structure of mental models, dubbed as “knowledge-as-theory” and “knowledge-as-elements”.

Knowledge-as-theory means that the mental model is “an overarching hierarchical conceptual structure with theory-like properties that constrains a student’s interpretation of subordinate models and ideas” (Özdemir & Clark 2007, p. 352).

“This theory-like knowledge is hypothesized to involve coherent structures grounded in persistent ontological and epistemological commitments. Because novices unconsciously develop these coherent structures through collections of daily experiences, their ‘theories’ are not available for hypothesis testing in a manner similar to scientists’ theories. However, novices’ alternative conceptions do constrain future learning and allow novices to make consistent predictions across conceptual domains. Knowledge-as-theory perspectives hypothesize revolutionary change in knowledge structures through various mechanisms.[...] [T]hey all assert that learners at any given time maintain a small number of well-developed coherent naïve theories based on their everyday experiences and that these theories have explanatory power to make consistent predictions and explanations across significant domains.” (Özdemir & Clark 2007, p. 354)

Conversely, knowledge-as-elements describes a structure in which “elements interact with each other in an emergent manner where the combinatorial complexity of the system constrains students’ interpretations of phenomena” (Özdemir & Clark 2007, p. 352). This corresponds to the quality of structure in the classification by de Jong & Ferguson-Hessler (1996), presented in section 3.1.1.

“[K]nowledge-as-elements perspectives hypothesize that naïve knowledge structures consist of multiple conceptual elements including, but not limited to, phenomenological primitives, facts, facets, narratives, concepts, and mental models at various stages of development and sophistication. Novices spontaneously connect

and activate these knowledge pieces according to the relevance of the situation. During the conceptual change process, the elements and interactions between the elements are revised and refined through addition, elimination, and reorganization to strengthen the network. From this perspective, conceptual change involves a piecemeal evolutionary process rather than a broad theory replacement process” (Özdemir & Clark 2007, p. 355).

### 3.2.4 Meaningful Learning

David Ausubel proposed a learning theory that deals with the differences between rote learning of information and *meaningful learning*. It also is fundamentally tied to concept mapping presented in the next chapter. A definition of a meaningful learning process can in short be given as follows:

“The essence of the meaningful learning process [...] is that new symbolically expressed ideas (the learning task) are related in a nonarbitrary, and nonverbatim fashion, to what the learner already knows (his cognitive structure in a particular subject-matter field), and that the product of this active and integrative interaction is the emergence of a new meaning reflecting the substantive and denotative nature of this interactive product.” (Ausubel 2000, p. 67f.)

Following the distinctions made for the mental process of rehearsal above, the theory of meaningful learning acknowledges that there is rote learning and meaningful learning as two different learning activities with different expected outcomes. The central idea of opposing modes of learning, with rote learning being one of them, can be found elsewhere as well. For example Biggs & Tang (2011) distinguish between “surface” and “deep” learning, with similar characteristics. Also, the learning process as observed by neurologists seems to validate the central idea of meaningful learning: “we only memorize what’s good and meaningful for us, learning is especially effective, when it makes sense. [...] we are learning by making associations and linking new information to current knowledge” (Sabitzer 2011, p. 168 f.). Meaningful learning results in deep structured knowledge, when following the classification by de Jong & Ferguson-Hessler (1996) presented in section 3.1.1: “‘Meaningful Learning’, by definition, involves the acquisition of new meanings. New meanings, conversely, are the end-products of meaningful learning. That is,

the emergence of new meanings in the learner reflects the prior operation and completion of a meaningful learning process” (Ausubel 2000, p. 67).

There are two prerequisites of a successful meaningful learning task present in the above definition, namely that the learners themselves choose to relate new information into their current cognitive structure in a nonarbitrary and nonverbatim fashion which in turn requires that the information presented to them is potentially meaningful, so that there actually is a way in which the information can be related to the learners’ cognitive structures. These two prerequisites are called *meaningful learning set* and *potential meaningfulness* (cf. Ausubel 2000, p. 68). To possess potential meaningfulness, information must be *logically meaningful*. This means that it must be “sufficiently nonarbitrary itself (i.e. nonrandom, plausible, sensible) so that it could be related on a nonarbitrary and nonverbatim basis to correspondingly relevant ideas that lie within the realm of what human beings are capable of learning” (Ausubel 2000, p. 69). Whereas the logical meaningfulness is an inherent property of the information and the way it is presented to a learner, the *potential meaningfulness* is a property of the learner, or more specifically, of the learner’s knowledge structure. If there is no way that an information can be related meaningfully into a certain cognitive structure, then the person that this structure belongs to will not be able to meaningfully learn this information regardless of whether the person chooses to learn meaningfully or not (cf. Ausubel 2000, p. 70).

Concerning the actual learning process, Ausubel (2000, p. 84ff.) distinguishes between three types of learning:

1. *Vocabulary or representational learning*, that describes “learning the meanings of single words, or learning what single words represent”.
2. *Concept learning*, which describes the acquisition of a new *concept* and can happen in two ways:
  - (a) *Concept formation*, in which “the criterial attributes of the concept are acquired as a consequence of direct experience through successive stages of hypothesis generation, testing, and generalization”.
  - (b) *Concept assimilation*, during which “the criterial attributes of new concepts can be ascertained by use in new combinations of existing referents (words as well as images) available in the child’s cognitive structure”.
3. *Propositional learning*, which describes a task that “consists of a composite idea and is expressed verbally in a sentence containing both denotative

and connotative word meanings and the syntactic functions of and relations between words”.

Concepts are defined as “objects, events, situations, or properties that possess common criterial attributes and are designated by the same sign or symbol” (Ausubel 2000, p. 88). Also, concept formation, according to (Ausubel 2000, p. 88) is mostly relevant for young children. For this work, only concept learning and propositional learning are relevant.

Each learning process, if done meaningfully, will integrate new ideas into an existing cognitive structure. This integration can follow certain hierarchical patterns (cf. Ausubel 2000, p. 89ff.) of which the following three are relevant for concept assimilation and propositional learning: *Subordinate learning* or *subsumption* describes the integration of new information under an already existing superordinate cognitive element of a cognitive structure. In contrast, *superordinate learning* occurs when several existing elements of a cognitive structure are subsumed under a newly learned idea. Finally, if neither of the two patterns is applicable for the result of a meaningful learning process, the learning is called *combinatorial learning*. A succession of subsumptive learning processes, i.e. starting with a general concept, is called *progressive differentiation*. The opposite, a succession of superordinate learning processes starting from a very specific concept, is called *integrative reconciliation* (cf. Ausubel 2000, p. 102).

The underlying idea of these learning processes, especially concept assimilation, is “that new meanings are acquired by the interaction of new, potentially meaningful ideas (knowledge) with previous learned concepts and propositions. This interactional process results in a modification of both the potential meaning of the new information and of the meaning of the concepts or propositions to which it is anchored, and also creates a new ideational product which constitutes its new meaning to the learner” (Ausubel 2000, p. 102). It is relevant for education, as “[c]lassroom or subject-matter learning is primarily concerned with the acquisition, retention, and use of large bodies of meaningful information such as facts, propositions, principles, and vocabulary in the various disciplines” (Ausubel 2000, p. 67). Since the information is meaningful, it should be learned as such. Additionally, Ausubel (2000, p. 77) points out the importance of meaningful learning in education “because it is the human mechanism par excellence for acquiring and storing the vast quantity of ideas and information represented by any field of knowledge”. Additionally, rote-learning doesn’t remediate misconceptions held by a learner (cf. Novak & Cañas 2010, p. 1), as it is by definition always an assimilation of knowledge and not a radical restructuring.

### 3.2.5 Models of Learning

Every learning theory typically has an underlying model of how persons actually learn. This model is not always explicitly stated, though. For example, the theory of meaningful learning is based around the model of a person who must be willing to integrate new material in a non-arbitrary and non-verbatim fashion. If this integration occurs, the person has learned (meaningfully). There are several explicit models of how a person learns, however. All of the models have in common that learning is seen as a personal change that occurs if a certain set of preconditions is met. Learning models are relevant since they can help explaining artifacts in the knowledge structures of persons, when monitoring an educational process, as shown later in the first case study in chapter 10. Also, many learning models stress the importance of existing prior knowledge. It is therefore paramount for teaching and learning, to take this into account, which requires identifying the existing knowledge.

Kolb & Fry (1975) and Kolb (1984) describe a basic model of learning that is based on the principles of Constructivism presented above - explicitly on the work of Lewin, Dewey and Piaget. According to this model, learning progresses through four stages as displayed in Fig. 3. It begins with a *concrete experience* of, for example, a phenomenon in the real world. This triggers a *reflective observation* of the encountered phenomenon. The result of this stage is an *abstract conceptualization* which in turn is refined and validated through *active experimentation* (cf. Kolb 1984, p. 39). The model is influenced by Constructivism since it emphasizes that the person is in charge of creating the knowledge subjectively and since the process of learning originated in a concrete experience. The four stages form the basis of four learning styles, that Kolb defines, based on a preference of a person for any of the different aspects of the learning cycle. As Hay, Kinchin & Lygo-Baker (2008, p. 296) state: "The approach subsumes the notions of difference, since it suggests that, while different people may have different affinities for one part or other of the cycle, ultimately learning occurs only when the cycle as a whole is complete."

Jarvis (1992) and more recently Jarvis (2012) presents a complex model of (adult) learning which emphasizes that learning is a personal endeavor that will result in a change of the learner, if successful.

Hay et al. (2008) integrate the ideas of the learning models of Kolb, Jarvis and the theory of meaningful learning into a more general model in order to measure the quality of learning. It is shown in Fig. 4. They identify several possible outcomes of learning that is visible in concept maps (see next chapter). Aside from the two extremes of rote learning and meaningful learning, there are also other possible developments of the knowledge. For example, rote learning can lead to the point



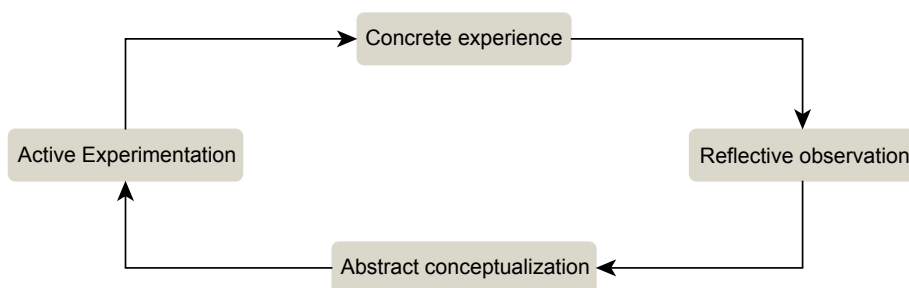


Fig. 3: Kolb's learning cycle

where new concepts cannot be integrated into the current knowledge structure, leading to a collapse and a subsequent re-emerging of the structure, in a different and (hopefully) more meaningful connection.

There are several other models of learning as well. For example, Wittrock (1992, p. 532) suggests a generative learning model in which “comprehension and understanding result from the processes of generating relations both among concepts and between experience of prior learning and new information. [...] This active generation process is quite different from the process of getting learners to store information for reproduction on lists”. The model, again, indicates the importance of existing knowledge.

### 3.3 Assessment

Learning is a process of changing the mind and the two processes of learning and organization of knowledge - as presented in the last two sections - are intricately related. Assessment is seen, in this work, in the specific context of education. It usually entails a scoring or grading of some externalized (see section 3.1.2) artifacts, like a response to a math problem, an essay, or a picture. Typically the person assessed is not assumed to be an expert. As has been mentioned before, the externalization of knowledge requires a form of representation. “How do we assess and represent an individual's knowledge? [...] The two processes, assessment and representation, are obviously related. In our view the approach to representation is more fundamental in that assumptions regarding the organization of knowledge have implications for how we assess knowledge” (Goldsmith & Johnson 1990, p. 241).

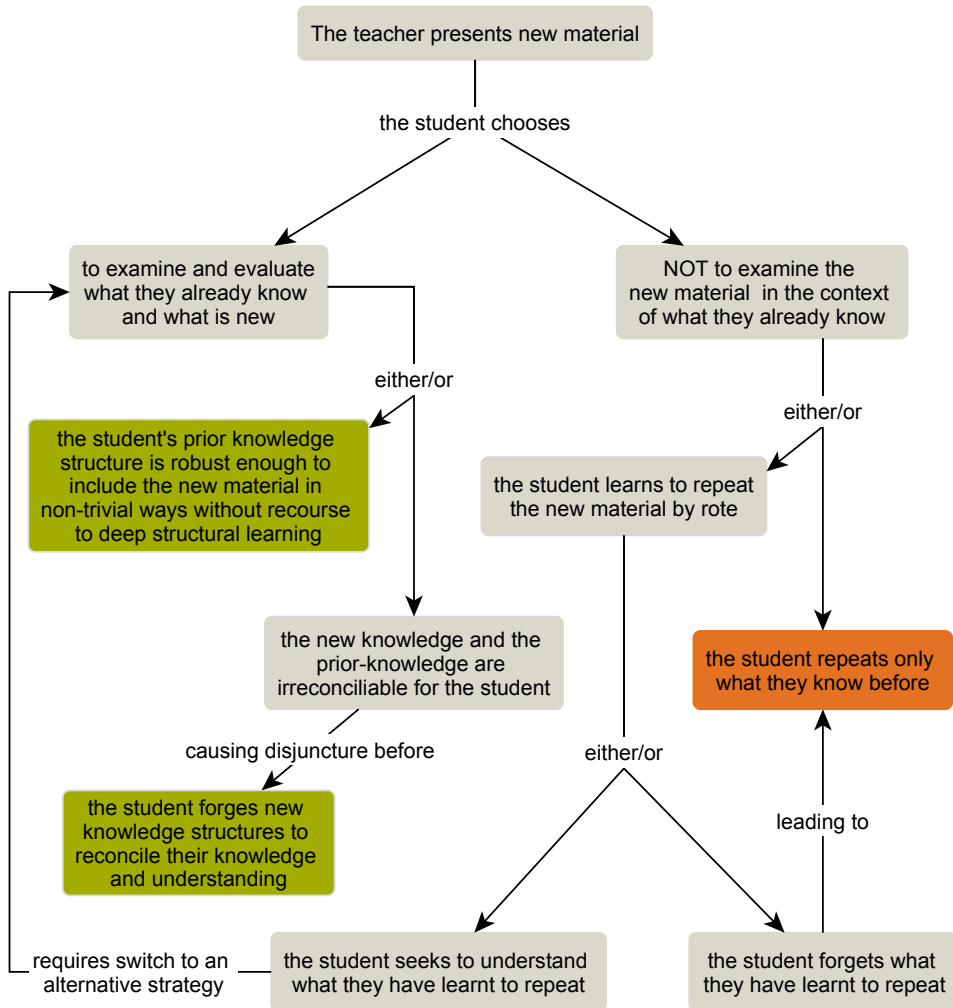


Fig. 4: A model of learning, adapted from Hay et al. (2008, Fig. 7). All boxes with no outgoing arrows represent final states of the model: The two green ones on the left represent an outcome in which the student has learned something. The red box on the right represents an outcome of non-learning.

An ideal assessment task “is objective and reliable, minimizes the influence of context on responses, and captures something of the structural nature of the subjects’ knowledge” (McClure, Sonak & Suen 1999, p. 476). Additionally, any method of assessing knowledge should both be *valid* and *reliable*. “Reliability is an expression of the proportion of the variation among scores that are due to object of measure. As variation due to error goes to zero, the reliability of an assessment goes to 1” (McClure et al. 1999, p. 477). Put differently: “A test is reliable when individuals with the same fund of knowledge obtain the same scores, or if a given individual obtains the same scores when the exam is repeated with no change of knowledge occurring between tests” (Novak 2010, p. 221). A valid test measures the construct that is supposed to measure (cf. McClure et al. 1999, p. 478).

The next two sections present two major ways of defining the desired results of an assessment. The first focuses on small, testable items that are specifically defined, the second puts more emphasis on complex interactions.

### 3.3.1 Learning Objectives

One way of characterizing a desired outcome that can be assessed is by formulating explicit *learning objectives*. A learning objective is comprised of a verb and a noun. “The verb generally describes the intended cognitive process. The noun generally describes the knowledge students are expected to acquire or construct” Anderson & Krathwohl (2001, p. 4f.). This definition is in accordance with others found previously in literature (cf. Anderson & Krathwohl 2001, p. 12). The noun and verb of the learning objective can be directly used in order to construct an assessment. For example the learning objective “the student remembers that the second world war ended in 1945” can be directly formulated into an assessment question: “when did the second world war end?”

#### 3.3.1.1 Taxonomies

When looking at different learning objectives, it is usually self-evident that some are more difficult to achieve than others. Partly, this difficulty arises from the specific element of knowledge that is part of each learning objective. But there is also a difference in difficulty arising from the cognitive process. Clearly, remembering that the second world war ended in 1945 is less complex than the learning objective “the student is able to divide two polynomials”. These two objectives differ in the cognitive function that is required (remembering a fact versus applying an algorithm)

and in the type of knowledge that is required (a single fact versus an algorithm or procedure).

Bloom (1956) was the first to identify a taxonomy of learning objectives by classifying the cognitive process into six categories: *Knowledge*, *Comprehension*, *Application*, *Analysis*, *Synthesis* and *Evaluation*. This taxonomy is “[o]ne of the most enduring and useful models” (Sousa 2009, p. 248). Especially, since Anderson & Krathwohl (2001) present a revised version of Bloom’s taxonomy of learning objectives. A second dimension is introduced that classifies the element of knowledge that is part of each learning objective. Also, to correspond better to the actual wording of a learning objective, the original dimension of Bloom has been rephrased into verbs and the last two categories were interchanged.

The *knowledge dimension* categorizes different types of knowledge appearing in learning objectives. The types are *factual-*, *conceptual-*, *procedural-*, and *meta-cognitive knowledge* (cf. Anderson & Krathwohl 2001, p. 27). The second dimension deals with the cognitive process that is needed for a learning objective. The possible cognitive processes are: *remember*, *understand*, *apply*, *analyze*, *evaluate* and *create* (cf. Anderson & Krathwohl 2001, p. 30).

Taking, for example, the learning objective “the student is able to sort a list of numbers using Quicksort”: In this case, “being able to sort” points to an application of knowledge, so the cognitive process dimension would be “apply”. The knowledge element is “Quicksort” which, arguably, is a form of “procedural knowledge”. Hence, this learning objective would be classified as “applying procedural knowledge”.

Particularly in the context of computer science, Fuller, Johnson, Ahoniemi, Cukierman, Hernán-Losada, Jackova, Lahtinen, Lewis, Thompson, Riedesel & Thompson (2007) argue that the taxonomy of Anderson and Krathwohl is lacking a distinction between more practically oriented cognitive processes and more theoretically oriented ones. They propose a three dimensional taxonomy by further splitting up the cognitive process dimension into a plane, with the categories *remember*, *understand*, *analyze*, and *evaluate* forming one dimension (called “Interpreting”). The second dimension (called “Producing”) is made up of the three levels: *none*, *apply*, and *create*. They continue to describe certain “pathways” through this two dimensional plane that are typical for certain approaches, like “trial and error” (cf. Fuller et al. 2007, p. 164).

### 3.3.2 Learning Outcomes

While the terms “learning objective” and “learning outcome” are sometimes used synonymously, in this work, a learning outcome is a more complex artifact that has been created as the result of an assessment task (or learning process) and has not been defined a priori by the assessor - in contrast to learning objectives. A learning outcome could thus be, for example, a report written by a student which is then assessed or analyzed. The next section presents a way of characterizing and classifying such an outcome with regard to the knowledge and abilities that went into its creation. Competencies, which are briefly described afterwards are taking this idea even further.

#### 3.3.2.1 SOLO Taxonomy

Biggs & Collis (1982) describe a taxonomy that - instead of classifying learning objectives based on their difficulty - focuses purely on the structure of the actual *learning outcome*. The taxonomy is concerned with reception learning of existing knowledge and is based on the theory of meaningful learning (cf. Biggs & Collis 1982, p. 3). It also puts more emphasis on the learner, instead of the learning material. “The difference, essentially, is that the Bloom levels are *a priori* ones, imposed in advance by the teacher, whereas we would prefer to use levels that arise ‘naturally’ in the understanding of the material” (Biggs & Collis 1982, p. 13).

Table 3.1 shows the taxonomy and some explanations of the levels. Capacity “refers to the amount of working memory, or attention span, that the different levels of SOLO require” (Biggs & Collis 1982, p. 26). The relating operation “refers to the way in which the cue and response interrelate” (Biggs & Collis 1982, p. 26). Additionally, Biggs & Collis (1982) also give the attribute of “Consistency and closure”, referring to the felt need of the learner to come to a conclusion that is consistent with the data and other possible conclusions (cf. Biggs & Collis 1982, p. 27), which increases with the levels of the taxonomy.

While learning objectives are formulated in a manner that allows direct testing of them and therefore also of the different levels of the taxonomy, using the SOLO taxonomy requires a different approach. The assessment must be designed in such a way to elicit a response that accurately captures the level of the respondent (cf. Biggs & Collis 1982, p. 177). Failure to do so will typically result in observed levels being too low. Higher levels of the SOLO taxonomy require elaborate reasoning and the inclusion of external facts of knowledge. Failure to accommodate for this

SOLO Level	Capacity	Relating operation
Prestructural	Minimal: cue and response confused	Denial, tautology, transduction. Bound to specifics.
Unistructural	Low: cue and one relevant datum	Can “generalize” only in terms of one aspect.
Multistructural	Medium: cue and isolated relevant data	Can “generalize” only in terms of a few limited and independent aspects.
Relational	High: cue and relevant data and interrelations	Induction: Can generalize within given or experienced context using related aspects.
Extended Abstract	Maximal: cue and relevant data and interrelations and hypotheses	Deduction and induction. Can generalize to situations not experienced.

Table 3.1: The levels of the SOLO taxonomy as described by Biggs & Collis (1982, Table 2.1)

fact in the design of an assessment will often lead to less elaborate answers than a respondent could produce.

Even though the focus of the SOLO taxonomy is on the response of a person, the assigned level is still not considered a personal attribute: “The SOLO taxonomy makes no attempt to infer a cognitive processing level although it might be argued that to perform at a relational level or an extended abstract level involves greater cognitive processing than that required for unistructural or multistructural since the learners not only have to be able to recall items, they have to show the relationship among items (relational) and draw conclusions (extended abstract)” Fuller et al. (2007, p. 155).

### 3.3.2.2 Competencies

Taking the idea of e.g. the SOLO taxonomy even further, a competency is a personal attribute that can be measured and that is based solely on an actual (observable) outcome. The idea behind competencies and competence based assessment is that “in a modern industrial society, education and professional qualifications can no longer be described according to a rigid canon of knowledge in specific subjects passed on from generation to generation. Instead building competencies has been identified as the main objective of education” (Klieme et al. 2008, p. 3).

In his seminal publication, McClelland (1973, p. 7) criticized a testing mentality that is focused on intelligence or general aptitude tests and suggested instead that “[i]f you want to test who will be a good policeman, go find out what a policeman does. Follow him around, make a list of his activities, and sample from that list in screening applicants.” There, already, one of the central ideas of competence based assessments is visible: Testing should be done as close to real-life scenarios as possible: “It seems wiser to abandon the search for pure ability factors and to select tests instead that are valid in the sense that scores on them change as the person grows in experience, wisdom, and ability to perform effectively on various tasks that life presents to him” (McClelland 1973, p. 8).

There is no single accepted definition of competence. Specifically for large scale assessments, competence can be defined as “context-specific cognitive dispositions that are acquired by learning and needed to successfully cope with certain situations or task in specific domains” (Klieme et al. 2008, p. 9). This, in turn, is based on a definition given by Weinert (2001, p. 27f.), which lists several components of competencies - among them knowledge, but also e.g. volition. By measuring competencies in authentic (though maybe simulated) real world tasks, a greater validity is reached than by purely assessing knowledge, which may be rote-learned (cf. Klieme et al. 2008, p. 9). “Novices can know a principle, or a rule, or a specialized vocabulary without knowing the conditions of effective application. In contrast, when experts access knowledge, it is functional or bound to conditions of applicability. [...] The progression from declarative knowledge to well-tuned functional knowledge is a significant dimension of developing competence” (Glaser & Bassok 1989, p. 635).

Education with the goal of acquiring competence must aim for the students to choose to learn meaningfully. “Beginners’ knowledge of a domain is spotty, consisting of isolated definitions and superficial understandings of central terms and concepts. As proficiency develops, these items become structured and are integrated with past organizations of knowledge” (Glaser & Bassok 1989, p. 647). It is obvious, though, that while declarative knowledge is in most cases a prerequisite of competent performance of complex skills, competence is not a direct consequence of declarative knowledge. For example the knowledge required to evaluate a mathematical proof and to generate one are nearly equivalent, but the execution of each skill is different (cf. Glaser & Bassok 1989, p. 653).

Assessing competencies is a non-trivial task. It involves defining a cognitive model of competencies, which is difficult because of the contextualized nature of competencies (cf. Klieme et al. 2008, p. 10). It also requires defining suitable psychometric models that relate the theoretical construct of competences to empirical assess-

ments (cf. Klieme et al. 2008, p. 12). These two have to be “translated into concrete empirical measurement procedures” (Klieme et al. 2008, p. 13). “Valid competence models basically serve two goals: they are used to define educational standards and to measure the development of competences in diverse settings” (Linck et al. 2013, p. 1). Currently, there exists only a limited number of theoretically defined competence models - for computer science, the *MoKoM* project tries to establish a model for certain aspects of the scientific field (cf. Linck et al. 2013).

Nevertheless, conceptual knowledge is, in general, an important component of competence. “Even in highly procedural domains, acquiring knowledge of concept relatedness may improve subsequent performance” (Trumpower & Goldsmith 2004, p. 443). The idea of competencies has influence on instruction however, insofar as “all investigators agree that useful knowledge is not acquired as a set of general propositions, but by active application during problem solving in the context of specific goals” (Glaser & Bassok 1989, p. 659). Additionally, differing skills require differing amounts of declarative knowledge which should be reflected in the teaching (cf. Glaser & Bassok 1989, p. 660).



## 4 | Concept Maps

Historically, *concept maps*<sup>1</sup> were invented in the 1970s as a tool to help structuring and visualizing the responses of children in clinical interviews (cf. Novak & Cañas 2010, p. 1). The interviews were part of a twelve-year research project. It investigated the premise, “that substantive learning of basic science concepts was possible — if quality instruction could be offered” (Novak & Musonda 1991, p. 118), by providing interactive lessons on audio tapes to first and second grade children. The researchers then used a technique based on Piaget’s clinical interviews (Piaget 1929) to monitor the change of the knowledge structures of the participants over the course of twelve years. To help evaluating the interviews and extract the information about relevant knowledge structures, a visual representation of the occurring science concepts and their connections was developed, which in the end became concept maps.

“Prior to development of the concept mapping technique, our analyses of conceptual change focused primarily on the kind and frequency of propositions made by interviewees before and after relevant instruction. We found this to be useful, but there was still a lack of clarity as to how concept meanings were related to one another over the whole domain of knowledge represented in the interview. The construction of concept maps permitted us to begin with the most general, most inclusive concept dealt with in the interview and to show propositional structures in a hierarchical arrangement, also illustrating important interrelationships among concepts included in different interviewee statements.” (Novak & Musonda 1991, p. 126)

Later on, the use of concept maps shifted from a specific technique for data analysis to a general technique for learning, teaching, and assessing structural knowledge. Usually, the learners create the concept maps themselves in these settings (cf. Novak & Cañas 2008, p. 5, p. 11ff.). The technique of concept mapping is fundamentally based on the ideas of Constructivism and meaningful learning (cf. Novak & Musonda 1991, p. 126). To this day, concept maps have been successfully used as learning and teaching aids as well as for the assessment and investigation of persons’ knowledge structures in countless scenarios, studies, and subject domains, see e.g. (Al-Kunifed & Wandersee 1990). Novak & Cañas (2010) present

---

<sup>1</sup>The term “concept map” has been trademarked, serial number 75230079 registered at the United States Patent and Trademark Office but abandoned in 1998

an in-depth review over relevant literature and many areas of application. Also, there is a bi-annual international conference solely dedicated to concept mapping<sup>2</sup>. Attempting to give an exhaustive review over the relevant literature is beyond the scope of this work. The subject domains to be found are ranging from human resource development (Daley, Conceicao, Mina, Altman, Baldor & Brown 2010), political science (Zimmaro, Zappe, Parkes & Suen 1999) over mathematics (Ozdemir 2005), biology (Kinchin 2000), training in dental medicine (Kinchin & Cabot 2009) and nursing (Akinsanya & Williams 2004), physics (Mistades 1999), didactics of informatics (Gouli 2007) to computer programming (Keppens & Hay 2008), computer science (Sanders, Boustedt, Eckerdal, McCartney, Moström, Thomas & Zander 2008), (Ertl & Mok 2010) and beyond. Also, concept maps have been used for other tasks in the context of education, like the planning (Novak & Cañas 2008) or comparison (Anohina-Naumeca, Graudina & Grundspenkis 2012) of curricula. The next sections present the related work concerning the definition of concept maps and their application in more detail.

## 4.1 Elements

Fig. 5 shows an example of a concept map, by Novak (2010). It contains three elements of interest: Above all, there is a *focus question*. Its purpose is to define a context and help focusing on relevant aspects more easily (cf. Novak & Cañas 2008, p. 11). The rest consists of graphical boxes that all contain a label and represent a *concept*. Concepts are defined as “perceived regularities or patterns in events or objects, or records of events or objects, designated by a label” (Novak 2010, p. 25). Two concepts that are linked by a connection are forming the basis of a *proposition*. “When two or more concepts are related by the use of what we will call linking words, propositions are formed. These become the fundamental units of meaning stored in our cognitive structure.” (Novak 2010, p. 26). A proposition is composed of the two concepts and the label of the connection itself. Originally, only those links that are meant to be “read” either horizontally or upwards should show an arrow-head (cf. Åhlberg 2004, p. 26). However, this is clearly not the case in the concept map shown in Fig. 5. Instead, all links without an arrow-head should be read from top to bottom (cf. Safayeni, Derbentseva & Cañas 2005, p. 743). To avoid misinterpretation, it is also customary to always use arrow-heads, though. The basic rules for creating a concept map are summarized by Hay & Kinchin (2006, p. 129) as follows:

---

<sup>2</sup>see <http://cmc.ihmc.us>

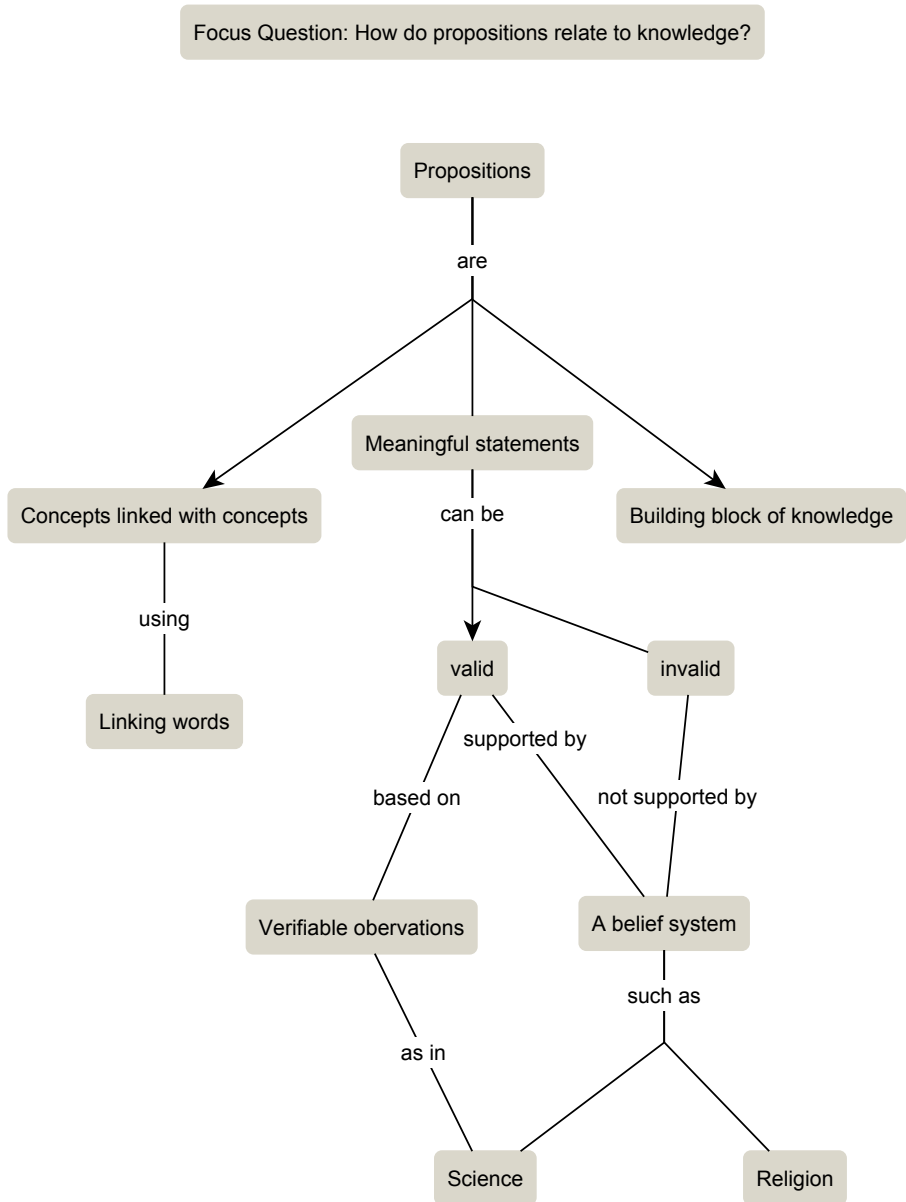


Fig. 5: A concept map illustrating the concept “proposition” and its relation to knowledge. Adapted from Novak (2010, p. 26)

- “The concepts that an individual deems important in illustrating their personal understanding of a topic are placed in text-boxes and arranged hierarchically on a page (so that broad and inclusive concepts are at the top and detail or illustrative example, at the bottom).”
- “Concepts are then linked with arrows that are annotated with “linking statements” to explain the nature of the link.”
- “Concepts may be listed only once, but any number of links may be made between any number of concepts at any number of conceptual links.”

A concept map, in its original intent is meant to answer a focus question (cf. Valerio, Leake & Cañas 2008, p. 122). Research has shown, that the formulation of the focus question has direct impact on the resulting maps’ structures (cf. Cañas & Novak 2012, p. 250). Also, it is necessary to define the context of the map (cf. Cañas, Carff, Hill, Carvalho, Arguedas, Eskridge, Lott & Carvajal 2005, p. 207), since concepts often have different meaning depending on the context, for example the concept “class” in the contexts of object orientation, biology, or education. However, when the context is clear to participants, the focus question is often more a task than an actual question that is meant to be answered by the map, which “usually leads to a descriptive concept map instead of an explanatory map” (Cañas & Novak 2012, p. 250). Omitting a specific focus question is especially appropriate for evaluating prior knowledge (cf. Cañas & Novak 2012, p. 251). Also, it is not necessarily seen as part of the concept map itself (cf. Cañas & Novak 2006, p. 496).

Safayeni et al. (2005) investigate the notion of concepts and propositions more closely. The results show that a concept in a concept map can stand for a psychological category (cf. Safayeni et al. 2005, p. 475 ff.). Also, it may be that certain, especially abstract concepts, “exist in the mind at the level of label with minimal description without references to any entity” (Safayeni et al. 2005, p. 748). Concerning the propositions, the following taxonomy of possible relationships between concepts and examples have been identified (cf. Safayeni et al. 2005, p. 480f.):

**Static Inclusion** a square is a geometric shape.

**Common membership** squares and triangles are geometric shapes.

**Intersection** squares have one more side than triangles.

**Dynamic Causality** travel time is an inverse function of the speed for a given distance.

**Correlation/Probability** academic performance in high school is a good predictor for academic performance at university.

Concept maps in their traditional form are typically not used to display dynamic relationships (cf. Safayeni et al. 2005, p. 481). Therefore, Miller & Cañas (2008, p. 2) slightly alter this definition by defining a dynamic proposition as involving “1) physical movement, 2) action, 3) change of state, or 4) [...] dependency or causal relationship”. Furthermore, dynamic propositions can be *causative* or *non-causative*: “In order for a dynamic proposition to be causative, one part of the proposition must embody the ‘cause’ or ‘probable cause’, while the other part must correspond to the ‘effect’. Alternatively, one part of the proposition must be identifiable as the ‘source’ from which that which the effect stated in the other part of the proposition originates” (Miller & Cañas 2008, p. 2). Causative propositions can also be quantified. “Quantified causative propositions explicitly indicate the manner in which a certain change in one concept induces a corresponding change in the other concept, unlike non-quantified propositions that make no reference to directionality or any other measure of the causal relationship” (Miller & Cañas 2008, p. 366). Based on this modification, both types of relationships can be expressed in concept maps, even though quantified dynamic relationships are usually found seldom (cf. Miller & Cañas 2008).

Most of the specific details of concept mapping as originally defined by Novak have been altered over time, allowing for a wide variety of concept mapping tasks to be found in literature - sometimes showing almost no similarity to the original (cf. Sousa 2009, p. 200ff.). Cañas et al. (2005, p. 208) define a “well-constructed” concept map as one where “[e]ach pair of concepts, together with their joining linking phrase, can be read as an individual statement or proposition that makes sense”, “[c]oncepts and linking phrases are as short as possible, possibly single words” and “[t]he structure is hierarchical and the root node of the map is a good representative of the topic of the map.” Similarly, Novak insists on his definition:

“Unlike so many ‘concept maps’ appearing in the literature. [*sic!*] what our team developed was a knowledge representation tool showing concepts and explicit prepositions forming a hierarchical structure. So-called concept maps that do not specify the links between ‘nodes’ fail to construct propositions which we see as the essential elements in representing meanings. The lack of hierarchy fails to indicate what concepts are most inclusive, or most salient for a given context to which the knowledge structure is to be applied.” (Novak 2002, p. 553)

### 4.1.1 Map Structure

The structure of a concept map is relevant, since it is supposed to reflect the knowledge structures of a person. As Trumpower et al. (2010, p. 5) note: “[K]nowledge organization has been recognized as important in the fields of education and educational assessment”. This is based on the premise that “knowledge requires not only acquiring facts, procedures, and concepts, but also having an understanding of the interrelationships among those facts, procedures and concepts” (Trumpower et al. 2010, p. 5). McClure et al. (1999, p. 491) assert that it is “the organizational component captured by concept maps that may allow teachers to identify and correct student misconceptions”. Koponen & Pehkonen (2010, p. 1670) have analyzed concept maps and conclude: “[W]ith more connections the structure also becomes more ordered. This suggests that students that are able to provide more connection (having more knowledge at their command) are also better at organizing that knowledge”.

A very basic structural aspect is small structural patterns between concepts. Novak (2010, p. 235) notes, without further elaboration, that “[o]nce nodes (concepts) have been placed in a map, they are related to one another to form larger graphic structures, usually triads”. Koponen & Pehkonen (2010, p. 1656) link several basic patterns, namely “different types of hierarchies, cliques, transitive patterns and cycles”, to observable artifacts of procedures of knowledge construction.

In contrast to these minimal structural elements of which a concept map is composed, it is also possible to investigate the overall structure of the map. Cañas & Novak (2012, p. 249) note that “[t]he overall structure of the concept map provides an idea of the global organization of the map, showing for example clusters of concepts in subdomains, whether the concept map is ‘balanced’ or whether one subdomain includes a much larger number of concepts and links than other subdomains”. As noted above, traditionally, a concept map is hierarchical. Propositions that connect non-adjacent levels of the hierarchy are called *cross-links*. Later on, cross-links are more generally described as “relationships or links between concepts in different segments or domains of the concept map” (Cañas & Novak 2012, p. 2). The hierarchy of a map becomes most important when concept maps are used as a teaching aid - alluding to the process of progressive differentiation: “Because meaningful learning proceeds most easily when new concepts or concept meanings are subsumed under broader, more inclusive concepts, concept maps should be hierarchical” (Novak & Gowin 1984, p. 15). In contrast, Ruiz-Primo, Shavelson & Schultz (1997, p. 7) investigate the necessity of a hierarchical structure and conclude that “[m]ethodologically and conceptually, there is no need to impose a

hierarchical structure on concept maps if the structure of the content domain to be mapped is not hierarchical. In fact, it may be that different map structures are needed to represent different types of content structures". They continue to point out, that there is no clear definition of a hierarchical concept map and that defining it - especially in the presence of cross-links - is a non-trivial task (cf. Ruiz-Primo et al. 1997, p. 8f.). One possible definition will be given later in this work. Åhlberg (2004, p. 27) suggests that in order to avoid ambiguities, concept maps should best be organized around a central concept, instead of forming a hierarchy. Safayeni et al. (2005, p. 745) note that the question whether or not a concept map must always be hierarchical is an open debate.

Leake, Maguitman & Reichherzer (2005) and Leake, Reichherzer, Cañas, Carvalho & Eskridge (2004) investigate elaborate ways of using a concept map's structure to determine the importance of single concepts for the map. Additionally, Hay & Kinchin (2006) present "conceptual typologies" of concept maps. According to the authors, concept maps can be divided into three major types of morphology:

**Spoke** One central concept acts as a "hub" and other concepts are connected directly to that "hub".

**Chain** The concepts are linked consecutively one to the other.

**Net** Several concepts form a (densely) connected network of propositions.

Fig. 6 shows an abstract example of the three different structures. Kinchin (2000) describes the process that led to the identification of these three types. They are the result of a research approach based on grounded theory and using different sets of concept maps created in the context of biology. However, e.g. Sanders et al. (2008, p. 333) report that using only these types of morphology didn't work for a study in object-oriented programming and Koul, Clariana & Salehi (2005) report a high variability in the correlation between qualitatively derived concept map scores based on these types of morphologies and scores of an essay writing task - in contrast to a quantitatively derived score. Yin, Vanides, Ruiz-Primo, Ayala & Shavelson (2005) suggest adding the types "tree" and "circular" to the list.

As has been mentioned at several instances in chapter 3, structure and organization of knowledge are assumed to be different between experts and novices. Therefore, it seems plausible to expect these differences to also show in the structure of concept maps. Table 4.1 presents differences between concept maps of experts and concept maps of novices that were found by Kinchin (2000) regarding the different types of morphology. Concerning the development of knowledge, the

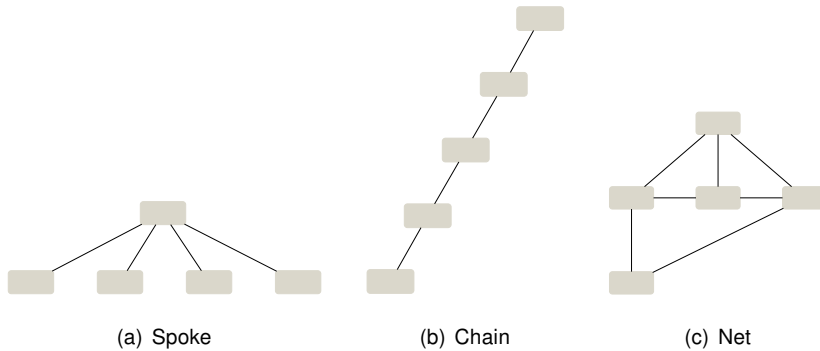


Fig. 6: The three different types of morphology of concept maps. Adapted from Kinchin (2000, p. 5-42)

structural developments visible in concept maps is closely tied to the learning model of Hay et al. (2008) shown in Fig. 4. Beginning with some form or prior knowledge, the rote- and non-learning outcomes are assumed to reflect by adding new concepts and propositions in a chain-like manner to the existing structure, whereas the meaningful learning is represented by adding new information into the knowledge structure, which makes the concept map more and more net like. A disjuncture that is caused by new facts which are irreconcilable with the existing knowledge structure manifests itself by a sudden collapse of the structure of the concept maps and a re-emerging of a more net-like map structure (cf. Hay et al. 2008, Fig. 6).

## 4.2 Applications

While there are many different applications of concept maps, the most prominent ones are arguably supporting teaching, learning and assessing structural knowledge. Other applications include the archiving of expert knowledge (cf. Cañas et al. 2005, p. 208) or supporting cooperative learning (cf. Mandl & Fischer 2000, p. 7).

“The widespread use of concept maps is based on the notion that a concept map is a reflection of the builder’s cognitive structure and thus portrays his or her understanding of the domain depicted in the map. For example, a concept map built by a student will show misconceptions as wells as concepts that are not clearly understood, and at the same time it makes evident what the student does understand.” (Cañas & Novak 2012, p. 247)



Characteristic	Expert	Novice
Connectedness	Highly integrated structure with numerous cross-links.	Disjointed structure dominated by linear arrangements in isolated clusters.
Link quality	Appropriate linking phrases which add to the meaning of concepts, using the specialist language of the domain.	Links are often inappropriate - usually single words that add little to the meaning and using non-specialist terminology.
Link variety	A diversity of linking phrases illustrating a range of thought processes.	The same linking words are used for a number of links, suggestive of a narrow range of thought processes.
Dynamism	Changes over time, reflecting active interaction with alternative knowledge structures.	Stable over time suggesting a lack of active engagement in knowledge restructuring.
Concepts	Concentration on overarching concepts to create an overview.	Concentration on specific concepts indicating a limited perspective.

Table 4.1 : Differences between concepts maps of experts and novices as identified by Kinchin (2000, Table 5-4)

When creating a concept map, there are four different possible scenarios. The simplest case is one person drawing a concept map at one point in time. Next, a person can continue to work on a concept map over a longer period of time, like creating a concept map over the course of a lecture by revisiting and changing the map over and over again. This approach has been called “recursive concept mapping” by Kern & Crippen (2008, p. 33). Alternatively, the concept map can be created collaboratively by several persons (cf. Chiu, Huang & Chang 2000), again at one instance or over a longer period of time.

### 4.2.1 Learning and Teaching

Since the initial use of concept mapping has been broadened to include teaching and learning, it has a long history of application in both settings.

“A concept map is a schematic device for representing a set of concept meanings embedded in a framework of propositions. Concept maps work to make clear to both students and teachers the small number of key ideas they must focus on for any specific learning task. A map can also provide a kind of visual road map showing some of the pathways we may take to connect meanings of concepts in propositions. After a learning task has been completed, concept maps provide a schematic summary of what has been learned.” (Novak & Gowin 1984, p. 15)

Concept maps are regarded as well suited for a constructivist approach to teaching and learning (cf. Kinchin, Hay & Adams 2000, p. 45). Especially for science education, Kinchin (2000, p. 63) remarks that in contrast to possible alternative graphical organizers “it is concept mapping [...] about which the literature has been so consistently positive”. Maps can be used to construct a shared meaning between persons, for example students and teachers, as well as serve to help students become aware of their own knowledge structures. Rye & Rubba (1998, p. 522) point out that concept mapping may foster meta-cognition in learners. Also, a concept map can serve as an *advance organizer* (cf. Ausubel 1968, p. 136f.) for students, helping them to integrate newly presented material into their personal cognitive structure. The usefulness of presenting knowledge to be learned in an organized way is generally accepted (cf. Goldstein & Vanhorn 2011, p. 178ff.). Additionally, Trumppower & Goldsmith (2004), Kinchin & Hay (2007), and Hay et al. (2008) point out, that instruction often requires the teacher to present in a linear fashion the

dense interrelations among concepts. Using concept maps can help students to see the original structure beyond the linear presentation. Novak & Gowin (1984), and Trumpower & Goldsmith (2004, p. 428) explain how the propositional structure of concept maps facilitates learning of new meanings, especially regarding concept assimilation:

“Except for a relatively small number of concepts acquired very early by children through a discovery learning process, most concept meanings are learned through the composite of propositions in which the concept to be acquired is embedded. Although concrete empirical props may facilitate concept learning, the regularity represented by the concept label is given additional meaning through propositional statements that include the concept.” (Novak & Gowin 1984, p. 15)

Nesbit & Adesope (2006) provide a meta-study that gives an overview over the effectiveness of concept maps as a learning aid. They also give a theoretical insight into why concept mapping may be effective. If viewing a concept map is akin to reading a map (i.e. a more visually oriented processing of the material), the authors argue that the dual coding theory might provide a basis for the effectiveness of concept maps. “Viewing or constructing concept maps in conjunction with semantically equivalent text or spoken presentations may facilitate cognitive representation of the information in both verbal and visuospatial memory. [...] Links between verbal and visuospatial codes provide additional retrieval paths for both types of information.” (Nesbit & Adesope 2006, p. 417). If viewing a concept map is seen more akin to reading a text, a different argument can be made for the effectiveness of concept maps when compared to text: “In maps, a concept is represented by a single node regardless of how many relationships it has with other concepts. That is, maps visually integrate propositions dealing with the same concept. In contrast, a concept may be represented at several places scattered throughout a text passage, and it may be represented by different words” (Nesbit & Adesope 2006, p. 418).

Regardless of the theoretical model underlying the effectiveness, the meta-study shows a clear trend toward the usefulness of concept maps when compared to other ways of presenting information like “activities such as reading text passages, attending lectures, and participating in class discussions” (Nesbit & Adesope 2006, p. 434). Also, when presenting preconstructed maps to students, it seems like there is a higher gain for students with lower reading abilities (cf. Nesbit & Adesope 2006, p. 434).

Karakuyu (2010) reports a measurable positive effect on achievement in and attitude towards physics when students constructed concept maps as part of the teaching. Kinchin (2000) also reports that the attitudes of teacher and students towards concept mapping is generally positive. "Concept mapping is a valuable tool that has enormous potential to support teaching and learning at all levels. Using concept maps should not be seen as an add-on activity [...], but as a core activity to stimulate the processing and synthesis of information" (Kinchin 2011).

### 4.2.2 Assessment

Using concept maps as a form of assessment is a natural extension of using them as teaching and learning aids. Kern & Crippen (2008, p. 33) explain how concept mapping can provide guidance for teachers in the planning of their lessons: "In a relatively short period of time, teachers can glean the following by viewing student concept maps: prior knowledge, misconceptions, and the acquisition and accommodation of new knowledge as maps are modified over time. The information derived from analyzing student concept maps can be used to tailor lessons to the immediate needs of students, resulting in a richer, more meaningful science learning experience".

The extent of what a concept map based assessment can cover is limited, though. "A concept map as an assessment can be thought of as a set of procedures used to measure important aspects of the structure/organization of a student's declarative knowledge" (Ruiz-Primo 2004, p. 555). Within these limits, however, Novak (2010, p. 231) and Novak & Cañas (2010, p. 3) point to the successful application of concept maps as a method for assessing or evaluating students' knowledge structures across many different fields of study. Especially, it has been shown that assessments based on concept mapping can differentiate between the knowledge of experts and novices as well as between meaningful learning and rote-learning (cf. Derbentseva, Safayeni & Cañas 2007, p. 450). Also, concept mapping may be especially useful in assessing misconceptions (cf. McClure et al. 1999, p. 476), (cf. Hay & Kinchin 2006, p. 130).

When used as an assessment task, the validity and reliability of concept mapping must necessarily be of concern. Establishing the reliability of an assessment task using concept maps is not easy: "[S]ources of error in a concept map test include: (a) variations in students' concept mapping proficiency, (b) variations in the content knowledge (domain expertise) of those evaluating the concept maps, and (c) the consistency with which the concept maps are evaluated" (McClure et al. 1999, p.

477). However, at least under certain conditions, the reliability has been established by several different studies (cf. Novak 2010, p. 232).

Concerning the validity, the results found in literature are generally positive: “The validity issue is relatively transparent since it is obvious that the fundamental characteristics of constructivist learning is exemplified in a well-constructed concept map” (Novak 2010, p. 231). Rosas & Kane (2012) draw a similar conclusion. Albert & Steiner (2005) present a more detailed overview over the problem regarding validity and suggest methods of determining it. One aspect that may interfere with validity is the meta-cognitive thinking that concept mapping fosters, as noted above. Going even further, Cañas et al. (2005, p. 208) establish that concept mapping is more than a passive externalization of an existing knowledge structure: “During concept map construction, meaning making occurs as the learner makes an effort to link the concepts to form propositions.” Leake, Maguitman & Cañas (2002, p. 168) comment that experts who are constructing a concept map “are not simply externalizing pre-existing internal knowledge but are also doing knowledge construction”. Additionally, Passmore (1999) provides a reasoning that integrates schema theory and concept maps, concluding, that concept maps can be representations of mental models. This is interesting in so far, as it provides a connection between concept mapping and the thinking process itself. Instead of seeing concept maps merely as an externalization of a person’s knowledge, it can also be seen as the representation of a mental model that existed in working memory during the thinking process of a person.

In general, every externalization will be influenced by many different variables which are neither completely known nor easily (or at all) measurable. The extent to which the knowledge of a person has been externalized in a concept map cannot be quantified. As Cañas & Novak (2012, p. 249) put it: “However, even when using the best rubrics that include both content and graphical structure, the educator needs to understand that the type and quality of the concept map may be more a reflection of the process and conditions under which the concept map was constructed than of the student’s understanding of the domain”.

Concerning the assessment task itself, Ruiz-Primo & Shavelson (1996) present the results of a qualitative analysis of a large body of literature in order to present and organize the multitudes of concept map assessment tasks. Their results show that concept mapping is interpreted and used in a highly variable way throughout literature. Their analysis results in the following classification system (the examples given are also taken from Ruiz-Primo & Shavelson (1996, p. 573ff.) but have been adapted):

### 1. Assessment task

- (a) task demands (e.g. filling in blanks in a concept map as opposed to constructing a complete concept map from scratch),
- (b) task constraints (e.g. being asked to construct a hierarchical map),
- (c) task content structure (e.g. the subject that is mapped has an inherently hierarchical structure).

### 2. Response format

- (a) response mode (e.g. drawing a map or taking part in an interview),
- (b) characteristics of the response format (e.g. a skeleton map is given on a sheet of paper),
- (c) mapper (e.g. a student as opposed to an interviewer).

### 3. Scoring system

The items of the classification systems have some overlap, i.e. are not fully independent from one another. For example, if the task consists of taking part in an interview, the response format will normally not be a concept map. Therefore, Ruiz-Primo (2004) mapped the complete classification system onto a one dimensional continuum of “Degree of Directedness”. Anohina-Naumeca & Graudina (2012) present a detailed discussion of the different possibilities of directedness. A very low degree for example is creating a concept map without any further restrictions and a very high degree of directedness is filling in blanks in a pre-constructed concept map (cf. Ruiz-Primo, Schultz, Li & Shavelson 2001). Ruiz-Primo (2004, p. 561) indicates that the decisions along the dimensions of the classification system are complex. “Which technique(s) should be considered the most appropriate for large scale assessment? Practical issues, though, cannot be the only criterion for selection. We have shown that the constraints and affordances imposed by different forms of assessments affect the student’s performance. This means that different mapping techniques may lead to different conclusions about students’ knowledge”.

The *task demands* are relevant, since they may influence the process of externalization. To support this, Ruiz-Primo et al. (2001) compare the results of a low directed “construct a concept map from scratch” task with a high directed “fill in the blanks in a concept map” task. The results show that the tasks, when compared to a multiple choice test, do not seem to measure the same aspects of student knowledge. Also, filling in blank concepts and filling in blank propositions do not seem to be equivalent assessment tasks. Yin et al. (2005) investigate the difference between choosing edge labels from a list of possible labels and freely creating edge labels without

restrictions. Again, the tasks seem to measure different aspects of a person's knowledge with the non-restricted task being better suited for capturing misconceptions and partial knowledge pointing to a higher validity (cf. Yin et al. 2005, p. 181). The restricted task however, showed a higher reliability concerning the scoring of propositions.

The very low directed task of creating a concept map from scratch without any *task constraints* has been referred to as the "gold standard" of concept map assessments (cf. Yin et al. 2005, p. 167). However, Cañas & Novak (2012) list several common restrictions, among which is providing a list of concepts that should be used for map construction. They note that "research has shown that the same students construct better maps when given a *list of concepts* [...]. More specifically, even if the number of concepts is similar in both cases (with the given list and without), the structure of the maps is different. Without the list of concepts the students tend to use one central concept and the maps have 'star' structure" (Cañas & Novak 2012, p. 5). Even more restrictive is an assessment task in which the participants of a concept map assessment may only use the concepts from that list, called *restricting list of concepts*. "Providing a restricting list of concepts is an effective way of determining the students' prior knowledge at the beginning of a study unit" (Cañas & Novak 2012, p. 251).

The *task content structure* of the subject matter underlying the task is assumed to reflect back upon the concept maps. It is therefore also an attribute of the assessment task. For example, if the content structure is hierarchical, one would expect concept maps to also reflect that hierarchy. If the content structure is more linear, a more linear map is to be expected from an assessment task (cf. Ruiz-Primo & Shavelson 1996, p. 578). However, Kinchin (2013, p. 100) points out that "whilst some structures are more or less contextually appropriate in a given situation, the student needs to appreciate this and to construct understanding accordingly". In other words, the replication of the content structure as agreed upon by experts in a concept mapping task may not be a feature of the assessment task, but more of the person being assessed.

Concerning the *response format*, mostly the *response mode* is interesting for this thesis, as it encompasses the use of pen and paper or software for concept map creation (cf. Ruiz-Primo & Shavelson 1996, p. 579f.). Using software to draw the concept map allows, for example, refining a map over time more easily than when using traditional methods. Kern & Crippen (2008, p. 33) believe that this fosters metacognition: "Metacognition, the act of thinking about one's own thinking, is a critical component in the conceptual change process. Recursive concept mapping, which involves building on and restructuring the same concept map over time,

can scaffold this change by providing an avenue for representing and evaluating students' thinking". Kwon & Cifuentes (2009) investigated the difference between digitally drawn concept maps and pen and paper based ones and concluded that digital creation seems to have a positive influence on students' motivation during the task. The next section investigates the last part of the classification in greater detail - the scoring of concept maps.

### 4.2.2.1 Scoring System

"A scoring system is a systematic method with which students' concept maps can be evaluated accurately and consistently. As expected, a myriad of alternative scoring systems can be found. However, they can be classified into three general scoring strategies: (a) score the components of the student's map, (b) compare the student's map with a criterion map and (c) use a combination of both strategies" (Ruiz-Primo & Shavelson 1996, p. 581f.). Besides the overview of Ruiz-Primo & Shavelson (1996), Anohina-Naumeca & Grundspenkis (2009) and more recently Strautmane (2012) also present a systematic evaluation of many different quantitative and qualitative scoring methods. So, instead of trying to give yet another overview, this section only presents general results concerning the quantitative and qualitative evaluation of concept maps as selected case studies of typical approaches. Whether scoring a map at all is a valid approach is not without doubts, especially when following a radical Constructivism. Novak & Gowin (1984, p. 97) point out, that "[c]oncept maps can be similar to paintings; you either like one or you do not". However, given a more moderate world view, Valerio et al. (2008, p. 122) state:

"Despite the variety of concept maps that arise from the differences among map builders, some maps can be considered 'better' than others, based on a variety of criteria.[...] Among the features that can be used to assess the quality of a concept map, we can distinguish between topological features (e.g. hierarchical structure, linking phrases, number links into and out of concepts, etc.), and semantic features (are the propositions correct? how expressive are the linking phrases? is the focus question answered by the concept map?)"

There are two general approaches that are apparent in the scoring schemes presented in literature. First, it is possible to treat the concept map as a mere accumulation of propositions that are scored independently. The concept map is then



only a graphical representation of that set of propositions. Alternatively, the concept map can be seen as one big “whole” that is scored holistically. Such a scoring can for example take the specific layout of the map into account, or the hierarchical structure, like the scoring system presented in the next paragraph. Structure and propositions are not always independent, though. For example, if the links have arrows, then usually only the specific wording of the linking phrase will determine the direction of the arrows, by using the active or passive form. Incorporating the structure of a concept map into its evaluation means abandoning the idea of the map being a mere accumulation of propositions and acknowledging instead that there is a structural aspect besides the propositional “content” of a map (cf. Cañas & Novak 2012, p. 248). Also, it is worth noticing, that the structural aspects of a map can always be judged, whereas the propositional content can only be judged if the map’s content is in some way “objective” and not, e.g. externalizing beliefs (cf. Cañas 2008).

Historically, the first scoring system for concept maps is most probably given by Novak & Gowin (1984). In contrast to many other scoring schemes, it is based directly on a learning theory, namely the theory of meaningful learning:

“The primary basis for our scoring schemes is Ausubel’s cognitive learning theory, especially three ideas in it [...]: (1) Cognitive structure is *hierarchically organized*, with more inclusive, more general concepts and propositions superordinate to less inclusive, more specific concepts and propositions. (2) Concepts in cognitive structure undergo *progressive differentiation*, wherein greater inclusiveness and greater specificity of regularities in objects or events are discerned and more propositional linkages with other related concepts are recognized. and (3) *Integrative reconciliation* occurs when two or more concepts are recognized as relatable in new propositional meanings and/or when conflicting meanings of concepts are resolved.” (Novak & Gowin 1984, p. 97)

(Novak & Gowin 1984, p. 107) give general rules of the scoring system as well as a more precisely defined, algorithmic approach (cf. Novak & Gowin 1984, p. 36):

1. *Propositions*: For each meaningful, valid proposition shown score 1 point.
2. *Hierarchy*: Is each subordinate concept more specific and less general than the concept drawn above it? Score 5 points for each valid level of the hierarchy.

3. *Cross links*: Is the relationship shown significant and valid? Score 10 points for each cross-link that is both valid and significant and 2 points for each cross-link that is valid but does not illustrate a synthesis between sets of related concepts or propositions. Unique or creative cross links might receive special recognition, or extra points.
4. *Examples*: Specific events or objects that are valid instances of those designated by the concept label can be scored 1 point each.

Following those four rules, a fifth rule states: "In addition, a criterion concept map may be constructed, and scored, for the material to be mapped, and the student scores divided by the criterion map score to give a percentage for comparison. (Some students may do better than the criterion and receive more than 100% on this basis)" (Novak & Gowin 1984, p. 36). This is noteworthy, since many later scoring systems employ the use of such a criterion map, often in order to increase reliability between several raters.

McClure et al. (1999) investigated the theoretical and practical issues of several scoring methods. They encompass a holistic approach, an approach that puts most emphasis on the structure of the concept maps and an approach that scores the maps' propositions ("relations") on their own. Each method was tested with and without a criterion ("master") map. All methods except for "structural with master map" show a significant concurrent validity when comparing the scores with the similarity of the concept maps to the master map using a measure of structural similarity presented in the next chapter (cf. McClure et al. 1999, p. 489). It can therefore also serve as a scoring method and is also used for analysis in this work, as presented in chapter 7.

The method "relational with master map", i.e. scoring each proposition on its own and using a criterion map was found to provide the highest reliability between several raters. Fig. 7 displays the method in form of a simple diagram. In this thesis, an adaption of this scoring scheme using zero, one or two points has been adopted and found to work very well in practice.

There are also more complex quantitative scores. For example, Ifenthaler (2006) presents a scoring strategy based on three "levels" called *SMD*. The first level, *Surface*, is simply the count of propositions in a concept map and should serve as a rough indicator of its complexity (cf. Ifenthaler 2006, p. 45f.). The next level, *Matching*, is the diameter of the concept map, when treated as a graph (as described in more detail in chapter 7). In other words: the maximum of the shortest paths between each pair of nodes. It should provide deeper insight into the structure of

the concept maps (cf. Ifenthaler 2006, p. 47). Finally, the last level, *Deep*, is the similarity of the concept map to a master map. The similarity is calculated using a measure first introduced by Tversky (1977). This level should provide an insight into the semantic contents of the concept map (cf. Ifenthaler 2006, p. 49). The SMD method is compared to several other scoring methods, like the ACMM method by O'Connor, Johnson & Khalil (2004) and found to be a reliable and valid way of scoring concept maps (cf. Ifenthaler 2006, p. 98). The SMD system is also, in contrast to those presented above, explicitly suited for computer aided or automated scoring.

In the same vein, Gouli, Gogoulou, Papanikolaou & Grigoriadou (2005b) present a scoring system that is fully automated. It identifies several types of errors based on an expert map and a numerical score is computed from this. To improve scoring in this system, the use of WordNet (see next chapter) for identifying synonyms has been suggested (cf. Kornilakis, Grigoriadou, Papanikolaou & Gouli 2004). Taricani & Clariana (2006) describe a method that converts a single concept map into relatedness ratings on which a Pathfinder analysis (see also next chapter) is applied. The same is done for an expert concept map and the resulting Pathfinder networks are then compared and scored automatically. This method is interesting insofar, as it departs from incorporating the actual propositions in the scoring and instead fully relies on the structural information contained in the concept maps. Anohina-Naumeca, Grundspenkis & Strautmane (2011) also present an elaborate automated scoring system that works in conjunction with an expert map.

From a constructivist perspective, using a referent concept map might be inherently problematic, as Cañas et al. (2005, p. 207) argue: "The strength of concept maps lies in their ability to measure a particular person's knowledge about a given topic in a specific context. Therefore, concept maps constructed by different persons on the same topic are necessarily different, as each represents its creator's personal knowledge. Similarly, we cannot refer to the correct concept map about a particular topic, as there can be many different representations of the topic that are correct."

So far, the scoring systems presented have all been quantitative. They typically assign some score to a map which then allows ranking several maps according to the score. However, it is worth noticing, that an overly quantitative approach is not universally accepted:

"The scoring of only 'valid links' also misses the point that 'invalid' links may have a value to the student by supporting more valid links (sometimes temporarily) and so contributing to the overall knowledge structure that he or she is using as a basis for further

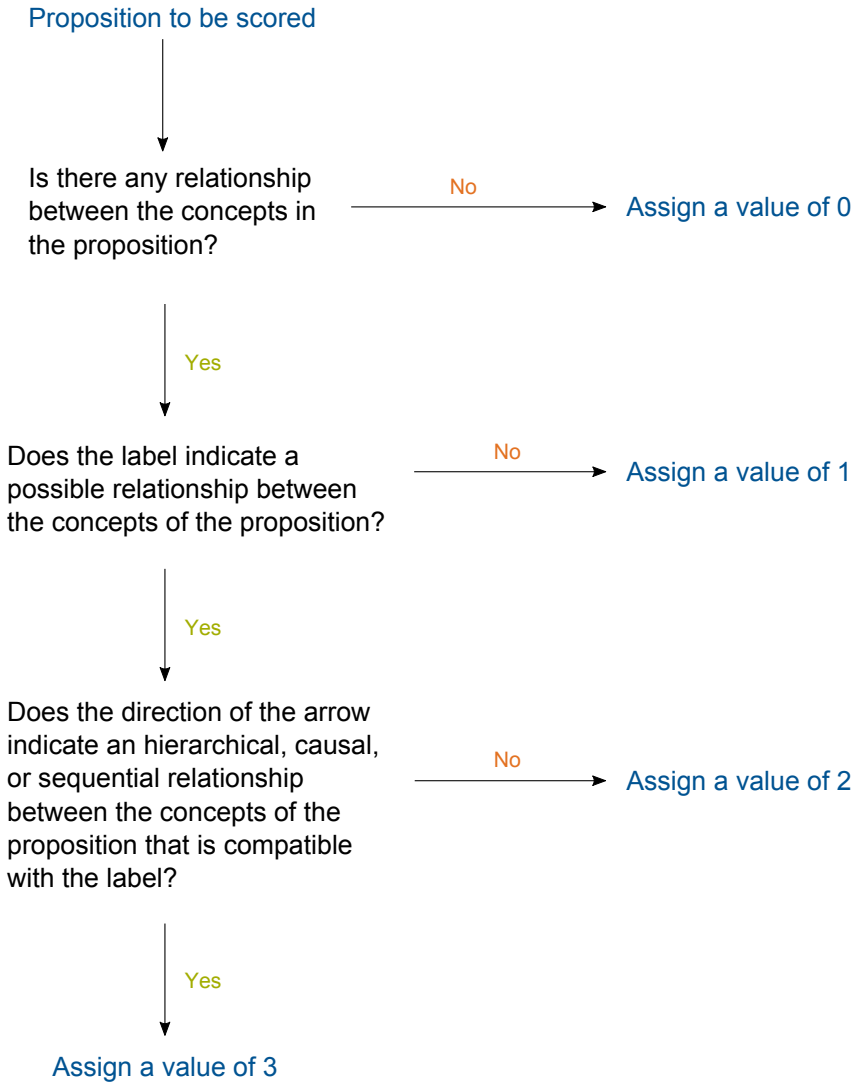


Fig. 7: Relational scoring method. Adapted from McClure et al. (1999, p. 482)

learning. The usual emphasis on 'valid links' seems to contradict the constructivist philosophy underlying the use of concept maps by failing to recognize the significance of students' perspectives. [...] This suggests that a more informative assessment of concept maps is required that could be used to bring benefits to the students' learning experience while not placing unrealistic demands on the classroom teacher. To satisfy these requirements, a more qualitative description may be appropriate." Kinchin et al. (2000, p. 46)

There are also other reports that support the idea of not scoring the correctness of propositions (cf. Taricani & Clariana 2006, p. 68). Nicoll (2001, p. 870ff.) presents a three-tier coding scheme that assigns attributes to propositions based on their "utility", "stability", and "complexity". This allows a more fine grained analysis of the propositions in a map, as they are not only classified with regard to their correctness, but also their type (e.g. "example") and regarding whether or not a student seems to be sure about the proposition or not.

Following their criticism of a purely quantitative approach, Kinchin et al. (2000) present a qualitative approach that classifies concept maps according to their "overall structure". The approach is based on the three types of morphology "spoke", "chain" and "net" as presented above. This approach, again, places no importance on the "correctness" of the propositions in a map. It also, in contrast to all methods described previously does not assign a score to a map. Consequently, it cannot be used to form a "rank" of concept maps. However, maps can be partitioned into classes of differing structure. "As 'invalid links' are seen as being of equal importance to 'valid links' (in terms of teacher-awareness), the time-consuming (and sometimes arbitrary) process of assessing the validity of links is avoided. [...] Implicit in this classification is the development of increasing integration of a conceptual framework from spoke structures towards net structures" (Kinchin et al. 2000, p. 46). The clustering described in chapter 7 is an automated way of forming such classes, with no prior definition of their structure.

Novak (2010, p. 234ff.) presents an evolved version of the scoring scheme presented above that places more emphasis on qualitative aspects and can be seen as a combination of quantitative and qualitative aspects of scoring. It is based on two separate parts: "The taxonomy dealing with general structure of the concept maps we call the topological taxonomy and the rubric dealing with the quality of meanings we call the 'semantic' rubric". The taxonomy is based on five different criteria: "concept recognition, presence of linking phrases, degree of

ramification, depth, and presence of cross-links". The criteria for the rubric are: "concept relevance and completeness, presence of dynamic propositions, number and quality of cross-links, and presence of cycles". Even though not completely left out, the correctness of the actual propositions is of considerable less importance than in the above, quantitative scoring schemes and is at least separated from the structural aspects of the maps.

## 5 | Analysis Methods

This chapter presents the theoretical background and related work concerning analysis methods that were applied in the course of this thesis. The methods are presented in chapter 7 and applied in practical settings in chapters 10 to 12. Additionally, the basics of graph theory are presented here as well, as they are a prerequisite of Pathfinder networks and are used in the next part as well. As has been noted in the first chapter, the analysis methods are inspired by the field of educational data mining, in particular the two categories “clustering” and “distillation of data for human judgment”. Cluster analysis obviously is a method for the first category, Pathfinder networks are employed for the second. Additionally, the automated analysis of texts is shortly presented in this chapter, which is used in the software described in more detail in chapter 8.

### 5.1 Graph Theory

Graphs are mathematical models consisting of *nodes* and *edges*. The exact details of the definitions vary according to source. The following notation is based on Schvaneveldt, Dearholt & Durso (1988, pp. 337-338) but similar definitions can be found in numerous other works on graph theory like (Balakrishnan & Ranganathan 2012).

A (directed) *Graph*  $G = (V, E)$  consists of a finite set of nodes (vertices)  $V$  and edges  $E$ , that form a subset of  $V \times V$ . If  $E = V \times V$ , the graph is said to be *complete*. Two nodes  $i$  and  $j$  are *adjacent* (connected by an edge) if  $(i, j) \in E$ . For any edge  $e = (i, j)$ , vertices  $i$  and  $j$  are said to be *incident* to edge  $e$ . For undirected graphs, the edges  $(i, j)$  and  $(j, i)$  are equivalent, often notated as  $\{i, j\}$ . The number of nodes and edges is denoted by  $n = |V|$  and  $m = |E|$  respectively.  $N_G(v)$  is the *neighborhood* of vertex  $v$  in graph  $G$ , i.e. the set of nodes that are adjacent to  $v$ . An *induced subgraph*  $G' = (V', E')$  of  $G = (V, E)$  is defined by a subset  $V' \subset V$  and letting  $E' = \{(a, b) \in E | a, b \in V'\}$ .

Nodes and edges can be *labeled*, edges can also be *weighted*. In this case the graph is also called labeled and/or weighted. This is modeled by label functions  $l_v : V \rightarrow L_v$  and  $l_e : E \rightarrow L_e$  for some sets of Labels  $L_v$  and  $L_e$ . Accordingly, a weight function  $w : E \rightarrow \mathbb{R}$  maps a real valued weight to each edge. Sometimes, the weights are given as a matrix  $W \in \mathbb{R}^{n \times n}$  with  $w_{ij}$  denoting the weight of edge  $(i, j)$  if this edge exists - usually it is set to  $\infty$  otherwise.

A set of distinct nodes  $v_1, v_2, \dots, v_k$  that are connected by edges  $(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)$  form a *path* of length  $k - 1$ . If  $v_1 = v_k$ , the path is also called a *circle*. If there is no path between two nodes  $v_i$  and  $v_j$ , the two nodes are in different *components* of the graph. A graph that contains no circles is a *forest* a forest with exactly one component is called a *tree*. For weighted graphs, the *distance*  $d_{ij}$  between two nodes  $v_i$  and  $v_j$  is the minimal combined edge weight of all edges along any path from  $v_i$  to  $v_j$  according to some metric. Often, the metric is simply the sum of the individual edge weights, but other metrics, like the maximum of the weights along a path may be used as well. If there is no path between  $v_i$  and  $v_j$ , the distance is set to infinity  $d_{ij} = \infty$ .

When implementing graphs in a computer, they are often stored as an *adjacency matrix*  $A \in \mathbb{B}^n$ , with  $a_{ij} = 1$  if edge  $(i, j)$  exists and  $a_{ij} = 0$  otherwise. Thus, for undirected graphs, the adjacency matrix is symmetrical. For weighted graphs, often the adjacency matrix is defined as  $a_{ij} = w_{ij}$  if edge  $(i, j)$  exists and  $a_{ij} = \infty$  otherwise.

## 5.2 Pathfinder Networks

*Pathfinder networks* as described by Schvaneveldt, Durso & Dearholt (1989) are graph based representation of the similarity (or dissimilarity) of entities. Originally, the data that is represented consists of pairwise similarity ratings given by persons, usually using a numeric scale. The similarity ratings can be modeled as a weighted, complete graph with each entity becoming a node and the weight of each edge is the similarity value of the pair of incident entities of that edge. Such a representation is called a “network”. Schvaneveldt et al. (1989, p. 252) note that “[a]s psychological models, networks entail the assumption that concepts and their relations can be represented by a structure consisting of nodes (concepts) and links (relations). Strength of relations are reflected by link weights and the intentional meaning of a concept is determined by its connections to other concepts”. Algorithmic methods can then be used to analyze such a network, or extract prominent features. The Pathfinder algorithm is one such method and an alternative is, for example, *multi-dimensional scaling* (MDS) developed by Kruskal (cf. Bartholomew, Steele, Moustaki & Galbraith 2008, p. 55ff.).

The next sections will present the Pathfinder algorithm that is used to create the Pathfinder network and explain how it can be used in the context of structural knowledge.



## 5.2.1 Construction

As described by Dearholt & Schvaneveldt (1990, p. 2ff.), a Pathfinder network is always constructed from an existing, non-negative weight matrix, which represents a weighted graph. The Pathfinder network is itself again a graph that is directed if and only if the input graph was directed. It consists of the same nodes and components as the input graph but of a subset of its edges, with their weights preserved. The edges are chosen such that the final network provides a path of minimal distance between each pair of nodes according to a special metric (called *Minkowski-* or *r-metric*) that is dependent on a parameter  $r > 0$ . The weight of a path consisting of edges  $e_1, e_2, \dots, e_k$  with weights  $w_1, w_2, \dots, w_k$  according to the  $r$ -metric is defined as:

$$(w_1^r + w_2^r + \dots + w_k^r)^{1/r}$$

For  $r = 1$  the  $r$ -metric defaults to the sum of the single edge weights, for  $r = 2$  it is the Euclidean distance and for  $r = \infty$  the path weight is the maximum edge weight along the path (cf. Dearholt & Schvaneveldt 1990, p. 3). These three values are representing highly used metrics and are called *Manhattan distance*, *Euclidean distance*, and *Chebyshev distance* respectively (cf. Han & Kamber 2010, p. 73).

Additionally, a Pathfinder network with  $n$  nodes is guaranteed to be  $q$ -triangular for  $q \in \{1, 2, \dots, n - 1\}$ . This means, that the weight of any edge  $(i, j)$  is less or equal than the weights, according to the chosen  $r$ -metric, of any path between  $i$  and  $j$  that is of length at most  $q$  (cf. Dearholt & Schvaneveldt 1990, p. 3). In other words, when ignoring paths longer than  $q$ , the triangle inequality holds for each pair of nodes in the graph. If  $q$  is set to the maximal value of  $n - 1$ , the (regular) triangle inequality always holds.

Following the description of Dearholt & Schvaneveldt (1990, p. 6), the Pathfinder Network  $PFNET(r, q)$  (with two parameters  $r$  and  $q$ ) of a weighted graph  $G = (V, E)$  is constructed like this:

1. Start with the graph  $G' = (V, E' = \emptyset)$ .
2. Order all edges of  $E$  non-decreasingly according to their weight.
3. Following this ordering, for each edge  $e = (i, j)$  add  $e$  to  $E'$  if and only if the path of minimal weight according to the  $r$ -metric with at most  $q$  nodes between nodes  $i$  and  $j$  is as least as big as  $w_{ij}$ . In other words, including  $e$  keeps  $G'$   $q$ -triangular.
4. Return  $G'$  as the Pathfinder network.

Since the edges in step 2 are added in non-decreasing weight,  $G'$  will remain  $q$ -triangular throughout the algorithm. For every edge  $(i, j)$  that is added to  $E'$ , no path of less weight between  $i$  and  $j$  can be found later on, as each edge added in a later step has a weight at least as big as the direct path already provided by  $e$ , thus keeping the triangle inequality valid.

A problem with the algorithm as described above is, that it allows for arbitrary inclusion of edges in case of ties. So, in general, there are several possible Pathfinder networks of a given graph. For interpretation, it would be preferable to have a unique network that is generated for a given graph and parameter values. There is an additional way of creating Pathfinder networks that doesn't suffer from this problem. The algorithm described by Dearholt & Schvaneveldt (1990, p. 7f.) is reminiscent of the matrix based shortest-path graph algorithms like the algorithm of Floyd-Warshall (cf. Cormen, Leiserson, Rivest & Stein 2001, p. 629ff.). It works as follows:

Let  $W$  be the  $n \times n$  non-negative weight matrix of a (possibly directed graph)  $G$ , where all weights  $w_{ii}$  are assumed to be 0 and the entries for non-existing edges are assumed to be  $\infty$ . Starting with  $W^1 = W$ , compute  $W^{i+1}$  like this:

$$w_{jk}^{i+1} = \min((w_{jm})^r + (w_{mk}^i)^r)^{(1/r)} \text{ for all } 1 \leq m \leq n.$$

Where  $r$ , as before, defines the metric that is used to calculate the node distances. For  $r = 1$  this simply defaults to the update rule of the shortest-path algorithm. Additionally, compute the distance matrix for at most  $i$  steps  $D^i$  according to the following rule:

$$d_{jk}^i = \min(w_{jk}^1, w_{jk}^2, \dots, w_{jk}^i)$$

If all weights are non-negative, this rule will ensure that always  $d_{jj}^i = 0$ . The algorithm first computes  $D^q$  according to the rules given above and then deletes all edges from  $G$  where  $w_{ij} \neq d_{ij}^q$ . The resulting graph is a unique superset of all possible Pathfinder networks of the input graph and itself a valid Pathfinder network. Thus, it includes every edge that does not explicit violate the  $q$ -triangularity and therefore may include more edges than strictly necessary. It makes interpretation and comparison of a graph more reliable, however. All Pathfinder networks in this thesis were constructed using this matrix algorithm to ensure that no edges are missing arbitrarily. The algorithm can be implemented straight-forward in time  $\Theta(qn^3)$  with space requirements of  $\Theta(n^2)$ .

There are other algorithms that generate the pathfinder network faster, especially for the often used case of  $q = n - 1$ , where the running time of the original algorithm degenerates to  $\Theta(n^4)$ . Guerrero-Bote, Zapico-Alonso, Espinosa-Calvo,

Gómez Crisóstomo & Moya-Anegón (2006) describe a faster implementation called “Binary Pathfinder”. Quirin, Cordón, Santamaría, Vargas-Quesada & Moya-Anegón (2008) present an algorithm with cubic running time that only works for  $q = n - 1$ . For the very special (but frequently used) case of  $q = n - 1$  and  $r = \infty$  Quirin, Cordón, Guerrero-Bote, Vargas-Quesada & Moya-Anegón (2008) present a fast alternative that works with minimal spanning trees. However, for small graphs (all graphs in this thesis are considered small), the running time of the straight-forward algorithm doesn't pose restrictions on the practicality.

## 5.2.2 Investigating Structural Knowledge

There are many studies that have successfully employed Pathfinder analysis on similarity data in order to gain insights into aspects of structural knowledge. Since this work is not concerned with similarity data, this section only presents some results regarding the general applicability of the method in the context of knowledge evaluation.

First, Pathfinder analysis has been shown to capture latent organizational traits of knowledge. Trumpower & Goldsmith (2004) found out that structuring learning material according to their similarity in a Pathfinder network (as opposed to, for example, alphabetical) enhances learning. Similarly, Durso & Coggins (1990) report enhanced capabilities of recalling words of a list when these are ordered according to their similarity in a Pathfinder network: “Pathfinder revealed some latent structure useful for predicting recall order beyond that which could be linearly predicted from the ratings” (Durso & Coggins 1990, p. 42). They also found that the structure of the network matches networks that were intuitively drawn by researchers (cf. Durso & Coggins 1990, p. 33). In the same vein, Pathfinder networks have also been used in automatic summarizing of texts (cf. Patil & Brazdil 2007), which can also be seen as a form of organizational trait of knowledge. This also indicates the independence of the Pathfinder method from the relatedness judgments that are typically used as input.

Second, Pathfinder analysis can be used to make a restructuring of knowledge visible. Dayton, Durso & Shepard (1990) investigated the difference between knowledge structures following a moment of insight when solving a puzzle: “We have shown that people who solve an insight problem have a much different knowledge organization than do people who do not solve it or who are unaware of the problem. We have also shown that the correct organization is not achieved merely by exposure to the relevant information, and we believe this is evidence that the difference

in structure between problem solvers and nonsolvers is due to a sudden and substantial shift in connections[...]" (Dayton et al. 1990, p. 277).

Finally, Pathfinder representations of structural knowledge have been used as a method of assessment and to identify differences between novices and experts. "In summary, there is some evidence that experts can be distinguished from novices based on their cognitive structures. Classifications based on Pathfinder was successful at uncovering the latent structure inherent in the empirical ratings. Thus, a comparison of experts with novices supplies some validation of the psychological utility of Pathfinder" (Durso & Coggins 1990, p. 40). Trumpower et al. (2010) describe a system for assessing structural knowledge based on similarity ratings, Pathfinder networks and comparison to expert networks. Goldsmith & Johnson (1990) report the results of a study that investigates the usefulness of Pathfinder networks to assess the knowledge of learners: "[It is based on] the idea that configural properties of representations reflect important characteristics of an individual's cognitive system. We further assume that these configural characteristics can be compared in network representations by employing a method for assessing structural similarity between graphs" (Goldsmith & Johnson 1990, p. 244). The similarity ratings of students were analyzed using both the established methods of MDS and Pathfinder analysis. The raw similarity ratings themselves performed well as a predictor for the final course grade. 37% of the variance associated with students' final grades are accounted for by the correlation coefficient between a student's similarity rating and that of the course instructor (cf. Goldsmith & Johnson 1990, p. 249). The question then was whether scaling techniques like MDS or Pathfinder could yield a better prediction than the raw similarity data itself. The study compares several indexes to the final course points of the students. The indexes under comparison are the raw proximity data, MDS applied on the raw data, the correlation between the graph-theoretic distances of the students' Pathfinder network and the instructor's network, and the  $C$  value (see below) of a student's and instructor's Pathfinder network as described in (Goldsmith & Davenport 1990).

"[T]aken together, these results imply that Pathfinder networks do indeed contain unique predictive variance over the proximity ratings and MDS, and that a configural assessment of networks is a better index for assessing network similarity than correlations. Apparently,  $C$  better reflects commonalities between structures that happen to be important in assessing knowledge. We assume that the characteristics common to a student's structure and instructor's structure that are predictive of knowledge attainment exist at a global or configural level within those representations. This, of

course, is exactly the type of information that C is assumed to be good at assessing.” (Goldsmith & Johnson 1990, p. 250)

Concerning the actual analysis, Goldsmith & Johnson (1990) have tested several measures of graph similarity. They conclude that the structural similarity can be measured by comparing the neighborhoods of all nodes between two given graphs. Specifically, let  $G_1 = (V, E_1)$  and  $G_2 = (V, E_2)$  be two graphs that share a common set of vertices  $V = \{v_1, v_2, \dots, v_n\}$ . Then the similarity of  $G_1$  and  $G_2$  can be calculated by summing the similarity of each neighborhood as follows:

$$C = \frac{1}{|V|} \sum_{v \in V} \frac{|N_{G_1}(v) \cap N_{G_2}(v)|}{|N_{G_1}(v) \cup N_{G_2}(v)|}$$

The value of  $C$  will vary between 0 and 1, where 1 denotes structural identity and 0 denotes a completely different structure. Note, that the fraction is undefined, if the union of the neighborhoods is empty, i.e. when both nodes are unconnected in both graphs. As they are then structurally identical, however, it is convenient to define the value of the fraction as 1 in this case.

Concerning the choice of parameters for the Pathfinder algorithm, in general, the higher the values the sparser the resulting network (cf. Cooke 1994, p. 832). However, the special context of structural knowledge has some implications. For example, Durso & Coggins (1990, p. 32) state:

“With an exponent of infinity, Pathfinder makes only ordinal assumptions about the data. The second parameter,  $q$ , is a restriction on the number of edges in a path that Pathfinder will use in deciding if two concepts are already connected. The sparsest PFNET will result when Pathfinder is permitted to consider paths of any length, that is when  $q$  is equal to one less than the number of nodes. The most dense graphs result when Pathfinder can only consider a path as consisting of two edges, that is  $q = 2$ . [...] Although decisions about the  $r$  parameter can be justified on measurement assumptions, the decision concerning  $q$  is more difficult. [...] Both when picking  $q$  and when picking the dimensionality, several factors, including the illuminating power of the solution, must be considered”.

Gammack (1990) quotes an example that serves to show the problem of higher  $q$  values:

"In a linked set of countries, Jamaica may be considered similar to Cuba with a strength of (say) 2 units, and Cuba may be considered similar to USSR, also with a strength of 2. Particularly under the dominance metric, Jamaica and USSR are very close in network terms, but such implied proximity is psychologically hardly very meaningful. For such multifaceted concepts, relatedness may be judged on numerous grounds, and this must be considered when interpreting a network. The criticism may not apply to all networks but is particularly relevant to those with semantic properties. [...] This may imply that values of  $q > 2$  should be treated with caution." (Gammack 1990, p. 222)

Additionally, Gammack (1990) critically investigated the stability of Pathfinder networks in the light of different methods of eliciting proximity data. The results show, that the Pathfinder networks, when analyzed statistically, show a relative high agreement, but differ considerably in visual structure. "Until closely examined, it seems disturbing that the networks are so structurally different, despite demonstrable statistical agreement. Whereas the effects of noise or task context presumably contribute to this phenomenon, there is nevertheless the assumption that the network represents only a slightly distorted form of a network structure actually existent in memory. Questioning this assumption suggests no reason to suppose direct correspondence between a network and a pre existing memory structure" (Gammack 1990, p. 222). However, the authors concede, that the effect may just be the result of the inherent problem that the process of externalization is influenced by external factors:

"[T]he assumption that this knowledge structure would be transparently elicitable by objective methods having no influence on underlying structure also looks questionable. Since the tasks were different in nature, it may have been unreasonable to expect anything else. Clearly the tasks have had an effect in that they have elicited different structures, suggesting that they may be addressing different aspects of the same knowledge. Knowledge itself is surely stable, but representations of it are not, despite their specific value in particular contexts" (Gammack 1990, p.224).

They conclude that "[t]he instability of representation is neither caused by, nor unique to Pathfinder [...]. Despite some surface variation, Pathfinder descriptions were found to be particularly useful in the context of knowledge elicitation where concise and meaningful representations of expert domain conception were reliably produced" (Gammack 1990, p. 226).

## 5.3 Cluster Analysis

“Cluster analysis or simply clustering is the process of partitioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters.” (Han & Kamber 2010, p. 444) The result of a cluster analysis is also called a clustering. Since every clustering is a partition, each element of the data is assigned to exactly one cluster. For the probabilistic approach presented below, the probability for each observation to belong to each cluster is calculated. Cluster analysis is a typical research method in exploratory analyses, as it identifies pattern in the data which can then be interpreted (cf. Bartholomew et al. 2008, p. 17).

Concerning the clustering method, a general distinction can be made between *model free* methods and *model based* (or *latent class*) methods. According to Han & Kamber (2010, p. 449f.), for model free methods a further distinction can be made between *partitioning*-, *hierarchical*-, *density based*- and *grid based methods*. The descriptions in the following sections are restricted to the methods that are actually applied in this thesis. Details about this selection are given in chapter 7.

### 5.3.1 Partitioning Methods

A partitioning method takes all observations and assigns each observation to exactly one cluster, thus forming a partition of the data. The number of clusters typically must be given as a parameter of the algorithm beforehand (cf. Han & Kamber 2010, p. 451). The assignment of observations to clusters is determined by the “distance” of the observations from each other according to some measure of distance or similarity. Typical choices for this measure are the  $L_1$  or  $L_2$  norms, which are identical to the Manhattan and Euclidean distance, which are in turn identical to the Minkowski distance for  $r = 1$  and  $r = 2$  as defined in the last section (cf. Han & Kamber 2010, p. 72) .

Two very commonly used clustering algorithms of the partitioning variety are *k-means* and *k-medoids* (cf. Han & Kamber 2010, p. 451). Both methods work in an iterative way, starting from a random first clustering to a locally optimal solution. Also, both methods compute the quality of a clustering (that is maximized by the algorithms) based on the sum of distances from all the observations of a cluster to its “center point”. The difference is, that this “center point” is one of the actual observations for k-medoids, and the mean of all observations of a cluster for k-means. This makes only sense if calculating the mean is a valid operation on the

data, i.e. if the data is at least interval-scaled (cf. Bortz 2005, p. 21), which is usually not the case for the data used in this work.

The k-medoids clustering is described by Kaufman & Rousseeuw (2005, p. 68ff.). The most common implementation given below is called *Partitioning Around Medoids (PAM)*. The input consists of a distance matrix of the observations and the number of clusters  $k$  that the algorithm should produce. The distance matrix for  $n$  observations  $D \in \mathbb{R}^{n \times n}$  contains for each pair of observations, the distance according to the chosen measure of similarity. After initializing the  $k$  medoids with random samples from the observations, the basic steps of PAM are:

- 1 For each observation  $o$  and each cluster  $c$ :
  - Calculate the distance between  $o$  and the medoid of  $c$ .
  - Assign  $o$  to the cluster with the minimal distance.
- 2 For each cluster  $c$  and each observation  $o$  belonging to  $c$ :
  - Calculate the total sum of distances from  $o$  to all other observations of  $c$ .
  - Choose the observation with the minimal sum as the new medoid of  $c$ .

This is repeated, until no further change in the assignments of observations to clusters (and hence also not in the chosen medoids) occurs. Therefore, the solution depends only on the initial choice of medoids and the input. For this reason, the algorithm is often run several times with different start values to increase the chance of finding the (globally) optimal solution.

Concerning the optimal number of clusters, additional indexes are often used to judge the quality of a clustering. Gordon (1999, p. 60ff.) presents and compares several indexes. Of these, a well working and easy to compute index is used in this thesis. It is called  $G1$ , also known as the *Calinski-Harabasz pseudo F-statistic*. Another well-known index is the so-called *Silhouette* (Rousseeuw 1987) of a clustering.  $G1$  is defined as:

$$G1(c) = \frac{\frac{B}{c-1}}{\frac{W}{n-c}}$$

$B$  denotes the total sum of squared distances between all pairs of observations not sharing the same cluster (“between-cluster”).  $W$  denotes the total sum of squared distances between all pairs of observations sharing the same cluster (“within-cluster”);  $c$  and  $n$  denote the number of clusters and the number of observations respectively. The index effectively measures how close together the observations



of a cluster are and how disperse the clusters are from another. A higher value indicates a “better” clustering in the sense, that the clusters are either more spread apart from each other or the observations within the clusters are very similar or both. So, the index can be calculated for several given clusterings with differing numbers of clusters and then the optimal value of  $c$  can be chosen.

A partitioning algorithm will produce a result in any case, even if the data itself is perfectly homogeneously distributed in the multidimensional space and shouldn't actually be clustered. However,  $G1$  and other indexes that can be used to judge the quality of a clustering are not defined for just one cluster (for  $G1$  the denominator will be zero in this case). Therefore it is paramount to analyze the data prior to clustering regarding its tendency to actually produce meaningful clusters. This tendency, in other words, is the “non-uniformity” of the data in the multidimensional space that it is sampled from. The Hopkins index is a well-known indicator for this non-randomness of data (cf. Han & Kamber 2010). It is calculated as follows:

$$H = \frac{\sum_{i=1}^n U_i}{\sum_{i=1}^n W_i + \sum_{i=1}^n U_i}$$

Given a data set  $D$  of  $k > n$  observations,  $n$  of those observations are sampled uniformly and randomly. Additionally,  $n$  artificial, new “observations” are created randomly and uniformly distributed in the same multidimensional space as  $D$ .  $W_i$  then denotes the minimal distance of the  $i$ -th sampled observation to any observation from  $D$  (except itself) and  $U_i$  is the minimal distance of the  $i$ -th generated “observation” to any point in  $D$ .  $H$  is a measure of how non-uniform the observations of  $D$  are in the underlying space, by comparing the actual minimal distances with those that would be expected in a uniform distribution of the observations. According to Lawson & Jurs (1990), if  $n$  is chosen to be 5% of the number of observations,  $H \geq 0.75$  indicates a non-randomness of the data set on a 90% confidence level.

### 5.3.2 Model Based Methods

Model based clustering differs from the methods of the last section, in that there is an explicit (mostly stochastic) model underlying the clustering that allows calculating a probability of an observation belonging to a cluster. Clustering is then no longer the task of assigning observations to clusters, instead, “the task of probabilistic model-based cluster analysis on a data set,  $D$ , is to find a set  $C$  of  $k$  probabilistic clusters such that  $P(D|C)$  is maximized” (Han & Kamber 2010, p. 503). In other words: to determine values for the parameters of the model, such that the conditional

probability of the model given the observations (also called *likelihood* of the model) is maximized. Typically, the model parameters cannot be determined analytically (cf. Han & Kamber 2010, p. 503f.). Often the *EM algorithm* as described by Dempster, Laird & Rubin (1977) is used to find a (local) maximum of the parameter space in an iterative way. The EM-Algorithm performs the following two steps consecutively and repeatedly, starting from an initial guess of parameters and given observations.

**E-Step** Calculate the expected value of the log-likelihood. That is, calculate the probability of the observations, given the current parameter values.

**M-Step** Calculate new parameter values that maximize this expected value, given the observations and current parameter values.

The EM algorithm converges to a local maximum (cf. Han & Kamber 2010, p. 508) so, typical stopping criteria are either a number of iterations or the amount of change of the parameters between iterations. If the number of clusters changes, the probability of the model also changes. This allows - in contrast to model free methods - to estimate the quality of the clustering directly based on the model. Particularly, the case of just one cluster can be estimated as well and taken as an indication of whether or not the data is actually suited to clustering. When just trying several different numbers of clusters and estimating the parameters, there is a risk of overfitting: Often, the models can be made to fit the data (nearly) perfectly, by introducing more clusters (parameters). To prevent this, instead of maximizing the likelihood of the model parameters and observations by itself, a different term is maximized, that includes a penalty for introducing too many model parameters (cf. Stibor 2008). In this work, the often used *AIC*, or *Akaike Information criterion* (Akaike 1974) will be used. It is defined as:

$$AIC(X, k) = 2k - 2L(X)$$

where  $k$  is the number of model parameters and  $L(X)$  is the log-likelihood of the observed data (given the model parameters). An alternative to AIC is, among others, BIC (Schwarz 1978).

## 5.4 Text Mining

In the course of this work, text mining approaches have been used in order to automatically detect salient concepts of a given text (e.g. a textbook of a lecture)

and propositions of a given set of concepts. Since automated analysis of texts is a vast topic of research, any in-depth description is beyond the scope of this work. Instead, only existing approaches that have been used for the software described in chapter 8 will be described here. There may be more advanced and better working methods available, though.

The basic approach of using computer support for extracting the most important concepts of a text has already been suggested, for example by Trumpower et al. (2010). For concepts, which are often equivalent to nouns, a simple measure of salience is the frequency of occurrence of the word in a text. While this often works for specific terms of a subject matter (e.g. “class” in a text about object-oriented programming), there are usually also many non subject-matter specific words like “figure”, “page”, “exercise” or “student” (for curricula, for example), that appear very frequently. Therefore, two possible approaches of a more useful measure of salience are described here:

The first uses a term weighting that offsets the relative frequency by some additional factor. The so called *term frequency inverse document frequency* ( $TF*IDF$ ) method (Rajaraman & Ullman 2011, p. 8) has been used, for example, to generate search queries from concept maps that are used as support when creating concept maps electronically (cf. Leake et al. 2004). It offsets the frequency of occurrence of a word by the frequency of occurrence of the same word in a set of documents (corpus).

$$TF * IDF = \frac{n_x}{n} \cdot \log \frac{N}{N_x}$$

Where  $n_x$  and  $n$  are the number of occurrences of word  $x$  in a document and the total number of words in this document.  $N$  and  $N_x$  are the total number of documents and the number of documents in which the word  $x$  appears respectively. Therefore, a salient word for a given document is a word, that appears much more frequently in this document, than in the others. Clearly, this measure depends on a well chosen corpus. To exclude word like “exercise” by this measure, the corpus should best be composed of textbooks and not of, e.g. newspaper articles.

Another possible approach of identifying salient words for a given text is the observation that such words usually are not spread out uniformly over the complete text, but instead follow a Poisson distribution, i.e. are more often found in specific parts of a text than in the rest, (cf. Montemurro & Zanette 2013).

### 5.4.1 Existing Software Solutions

Two software solutions that have been employed in the course of this work are the *natural language processing toolkit (NLP)*<sup>1</sup> and *WordNet*<sup>2</sup> as described by in Miller (1995) and Fellbaum (1999).

The NLP toolkit is providing, among others, a *tokenizer* that splits an input text into word tokens and a *part-of-speech tagger* (cf. Toutanova, Klein, Manning & Singer 2003), that determines for several languages (only English and German are relevant for this work) the grammatical role that each word has in a sentence. This is especially useful for identifying nouns which are candidates for salient concepts. The tagger distinguishes 36 different functions of the *Penn Treebank Tag-Set* (cf. Mitchell, Santorini & Marcinkiewicz 1993) including, among others:

- “noun, common, singular or mass”,
- “noun, proper, singular ”,
- “noun, proper, plural”, and
- “noun, common, plural ”.

WordNet is a semantic network of the English language. However, as it describes a universal modelling structure for languages, there are also WordNets for other languages. All entries in the WordNet database consist of a form and sense. Where form is “a string over a finite alphabet, and a sense [...] is an element from a given set of meanings” (Miller 1995, p. 39). “Each form with a sense in a language is called a *word* in that language. A *dictionary* is an alphabetical list of words. A word that has more than one sense is *polysemous*; two words that share at least one sense in common are said to be *synonymous*” (Miller 1995, p.39).

The entities (words) of the semantic network are classified into the categories: *noun*, *verb*, *adjective* and *adverb* (cf. Miller 1995, p. 40). Between those entities, the possible semantic relations in WordNet are given in Table 5.1.

The basic relation of WordNet is Synonymy, as the word senses are represented by a set of synonyms of a given word form (a so called *synset*) (cf. Miller 1995, p. 40). It can also be used to reduce words to a baseform and therefore help to detect, for example, that a noun in singular and a noun in plural are actually referring to the same concept, even though the actual words are different. WordNet has,

<sup>1</sup><http://nlp.stanford.edu/software>

<sup>2</sup>Registered Trademark of Princeton University, see <http://wordnet.princeton.edu>

Semantic Relation	Possible between	Example
<i>Synonymy</i> (similar)	Noun, Verb, Adjective, Adverb	pipe - tube
<i>Antonymy</i> (opposite)	Noun, Verb, Adjective, Adverb	wet - dry
<i>Hyponymy</i> (subordinate)	Noun	maple - tree
<i>Meronymy</i> (part)	Noun	ship - fleet
<i>Troponymy</i> (manner)	Verb	march - walk
<i>Entailment</i>	Verb	drive - ride

Table 5.1: The possible semantic relations in WordNet (cf. Miller 1995, Table 1).

among many other applications, successfully been used for automated support of software-based concept map drawing (cf. Cañas, Valerio, Lalinde-Pulido, Carvalho & Arguedas 2003).



## **Part III**

---

# **From Concept Maps to Concept Landscapes**





# 6 | Possibilities and Limitations of Concept Maps

Chapter 4 has introduced concept maps as a tool for (among others) learning, teaching, and assessing. Also, several different aspects of knowledge organization, learning, and assessing have been presented in chapter 3. Fig. 8 integrates these different parts and shows how they are related in the context of this thesis. Concept maps are presented in more detail in this chapter. Based on the relevant literature of the last part, a detailed analysis and discussion of the foundations of concept maps in the light of everything presented in the chapters 3 and 4 is performed. Care is taken to clearly distinguish between results that have been taken from literature and own conclusions based on these findings.

The following sections will present three different views on concept mapping: First, it can be seen as a (psychological) instrument that externalizes structural knowledge. Questions regarding psychological processes and influences on the externalization are important in this view. Second, it can be seen as a mere tool of displaying knowledge. The restrictions of concept maps regarding the encoding of complex facts of knowledge must be taken into account then. Finally, third, it can be seen as an educational assessment tool. This view in a way reconciles the limitations defined by the previous two views. Finding reliable scoring methods and analyzing how artifacts of learning and teaching can be measured by concept maps are pivotal, for the educational view. The selection of these views and the aspects they focus on is based on the specific context of this thesis and its use of concept mapping. Clearly, there are other views on concept mappings as well.

## 6.1 Cognitive View

This section investigates the types of memory a concept mapping task can externalize and the psychological foundations of concept maps. This includes observations made on the process of concept map creation from data collected in the course of this thesis.

Following the distinctions made in section 3.1.1 concerning the different types of knowledge, it is clear that concept maps can be seen as an externalization of declarative knowledge as it has been “declared” by the concept mapper (cf. Ruiz-Primo & Shavelson 1996, p. 570) - or at least in the context of a concept

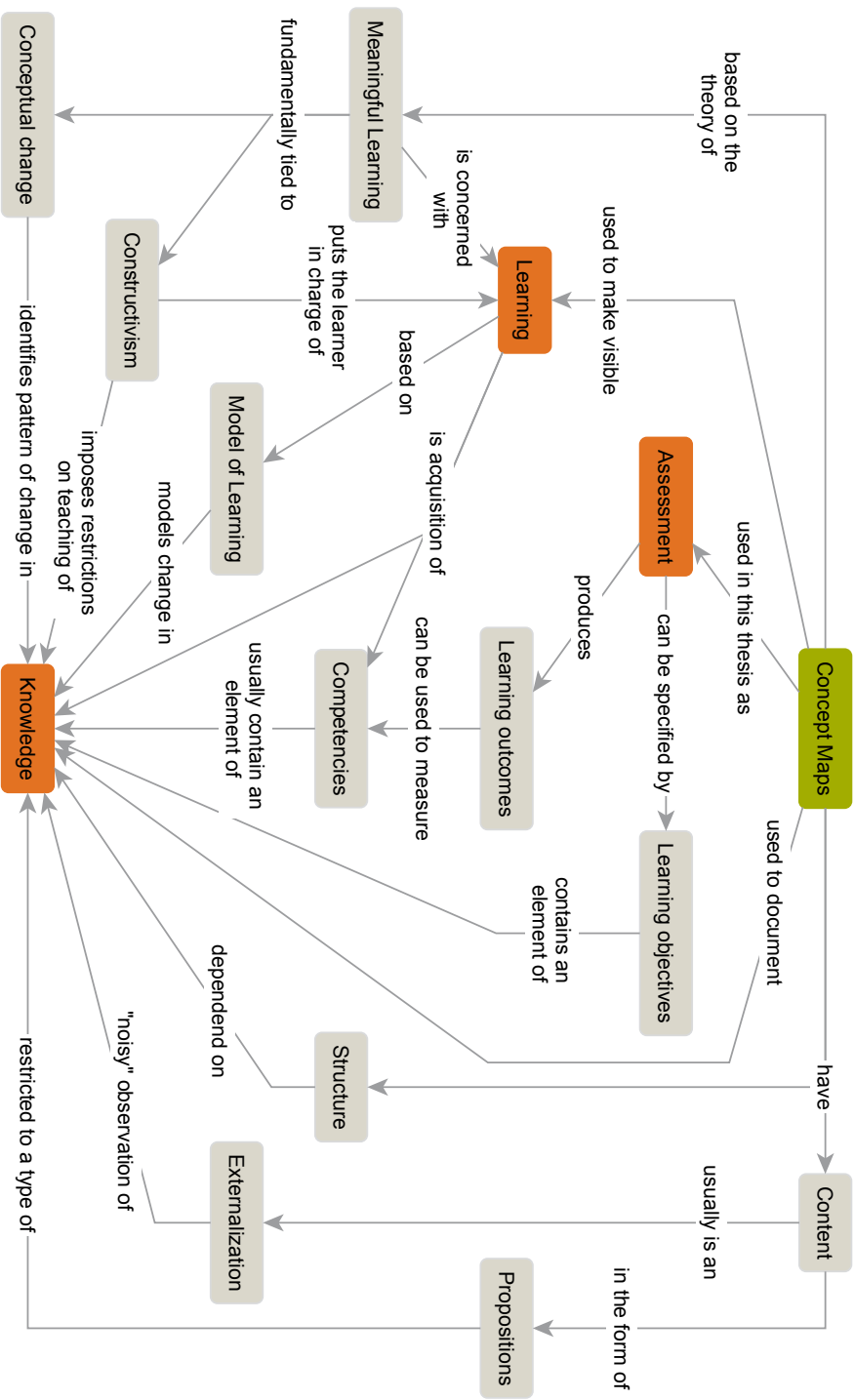


Fig. 8: Schematic overview over the interconnections of the topics of chapter 3 and chapter 4 in the context of concept maps.

mapping task. Thus, a concept map, in general, is an externalization of parts of the declarative, semantic memory of a person. As has been mentioned in chapter 4, concept maps are not restricted to knowledge or semantic memory, as it is also possible to externalize beliefs or personal experiences. Concerning the types of knowledge identified by de Jong & Ferguson-Hessler (1996) that have been presented in section 3.1.1, evidently, concept maps are suited to measure conceptual knowledge. They might also be able to measure situational knowledge, especially when taking into account that examples can be included in concept maps (see below). Procedural knowledge “contains actions or manipulations” (de Jong & Ferguson-Hessler 1996, p. 107). Since neither is an aspect of concept map creation - except for the procedural knowledge about concept mapping itself - it can therefore not be measured with concept maps. Strategic knowledge could in theory be measured by a concept map, however, as strategic knowledge “is applicable to a wider variety of types of problems” and concept mapping tasks are typically bound to a subject domain (cf. Ruiz-Primo & Shavelson 1996, p. 570), measuring it does probably not capture aspects of interest concerning the knowledge structures relevant for a subject domain. Regarding the qualities of knowledge identified by de Jong & Ferguson-Hessler (1996) also presented before, the following can be said for concept mapping:

**Modality** Since concepts and propositions are labeled with text, the modality must be verbal.

**Automation** Based on the definition of compiled knowledge given in chapter 3, the knowledge expressible in concept maps must be declarative.

**Generality** Given a subject domain, the generality is typically domain specific.

**Level** Both surface and deep knowledge can be externalized with concept maps, evidently, as a proposition expressed in a map can have been learned meaningfully or not.

**Structure** Both isolated elements and structured knowledge can be expressed in concept maps, which is directly visible in the maps structure.

Glaser & Bassok (1989, p. 35) give an explanation about why concept mapping may be inherently problematic when used for externalizing the knowledge of experts: “Experts and novices may be equally competent at recalling specific items of information, but experts chunk these items in memory in cause and effect sequences that relate to the goals and sub-goals of problem solution and use this information for further action. The progression from declarative knowledge to well-tuned functional

knowledge is a significant dimension of developing competence.” In other words, the particular type of knowledge organization in which experts differ from novices may not be adequately expressible with concept maps. The compilation of knowledge or, more general, the “chunking” of information that occurs with learning, as e.g. proposed by schema theory, may present a natural impediment to the assessment of a person’s knowledge structure, as often the chunks are transformed in such a way that makes explicit declaration of its “contents” difficult.

Next, the two central elements of concept maps namely concepts and propositions are investigated more closely. In the last part, there were definitions given in the context of concept mapping in section 4.1 and in chapter 3 both for the theory of meaningful learning (section 3.2.4) and concerning the knowledge organization in the brain (section 3.1.1):

**Ausubel** “objects, events, situations, or properties that possess common criterial attributes and are designated by the same sign or symbol” (Ausubel 2000, p. 88).

**Novak** “perceived regularities or patterns in events or objects, or records of events or objects, designated by a label” (Novak 2010, p. 25)

**Goldstein** “a mental representation that is used for a variety of cognitive functions, including memory, reasoning, and using and understanding language.” Goldstein & Vanhorn (2011, p. 240)

**Solomon** “the building blocks of thought” Solomon et al. (1999, p. 99).

The definitions of Novak and Ausubel are more oriented towards epistemology than the ones of Solomon or Goldstein. But, as Goldstone & Kersten (2003, p. 600) argue, both are valid in their own right: “If one assumes the primacy of external categories of entities, then one will tend to view concept learning as the enterprise of inductively creating mental structures that predict these categories. [...] If one assumes the primacy of internal mental concepts, then one tends to view external categories as the end product of applying these internal concepts to observed entities.”

Differences between both definitions exist, however: For example, Novak explicitly states the inclusion of “specific examples of events or objects” (Novak & Cañas 2008, p. 2) for clarification, which is in accordance with the second part of his concept definition but not necessarily in accordance with the definition of Goldstein. Whether or not a specific example is an actual concept depends on the way that this particular element of knowledge is stored in memory. It may be a rote learned fact of e.g.

a person's name or it may be a more elaborate association of this name with some relevant information. In the latter case, there may indeed exist a concept for that particular person while in the former case the person may just be a concrete entity placed in some category (e.g. "Politician"). As has been noted in section 4.1, concepts in concept maps have also been linked to psychological categories. Taking the example of "dog" (cf. Goldstone & Kersten 2003, p. 600), which can be both a concept and a category (subsuming all the entities of dogs encountered), using "dog" in a concept map may refer to:

1. the *mental idea* of a dog (i.e. the concept) as expressed by a proposition like "a dog has a tail", or
2. a subsumption of different *examples* of dogs (i.e. the category) as expressed by propositions like "Lassie is a dog".

Very often, concept mapping seems to be based on the notion that a concept is a technical term of the subject matter. This is the case for example when using "instance" as a concept in computer science (cf. Sanders et al. 2008), "double helix structure" (cf. Kinchin 2011) or "fermentation" (cf. Passmore 1999) in biology, "anatomy" in dental medicine (cf. Kinchin 2013), "momentum" in physics (cf. İngeç 2009), or even a formula (cf. Koponen & Pehkonen 2010). In general, this is not in accordance with the definition of Novak, as a technical term like "momentum" is neither an event nor an object. It can be in accordance with the psychological definition, as there can be a mental representation of the abstract notion of "momentum" for example. It can also be a simple rote-learned name or label and therefore clearly not a concept. In summary, a concept map consists of concepts that can be:

1. a (psychological) *concept*,
2. a (psychological) *category*, or
3. a rote-learned *fact* (label)

Propositions or associations - the connections between concepts - can then be seen as the result of the mental process of integration (see section 3.1.1). Integration finds connections between two concepts, which is exactly what a proposition in a concept map constitutes. In contrast to the epistemological view presented below, the restriction of propositions to be between two concepts only, is psychologically valid.

In conclusion, a concept map can externalize the results of the mental process of integration and displays the interconnections of a set of concepts. Goldstone

& Kersten (2003, p. 615) provide a reasoning about the validity of this approach beyond concept mapping: “[I]t is likely that all of our concepts are embedded in a network in which each concept’s meaning depends on other concepts as well as on perceptual processes and linguistic labels. The proper level of analysis may not be individual concepts, as many researchers have assumed, but systems of concepts”. Clearly, the concept mapping task itself may influence the results. For example, presenting a list of concepts may help retrieving an isolated fact from memory without the corresponding concept actually being a (psychological) concept for the mapper. Conversely, presenting a list of labels to be used as linking words (i.e. the labels of connections) can foster the creation of propositions that would not have been created without the provided list (cf. Yin et al. 2005, p. 177), or even the arbitrary creation of propositions that are not retrieved from memory but simply guessed.

Due to the nature of concept mapping, it is paramount to acknowledge that in contrast to other forms of externalizations, like relatedness judgments, for which missing links may be interpreted (cf. Rye & Rubba 1998, p. 543), it is in general not possible to deduct anything from missing elements in a concept map. It merely represents an externalized part of a person’s knowledge. This externalization process, as has been mentioned in section 3.1.2, can be influenced by a number of variables for example the volition of the person, the prior experience with concept mapping, the time given to create the map and more. If concepts or propositions are not contained in a concept map, the knowledge structure of the person might nevertheless contain them. This is not generally followed, though, for example the scoring system presented by Gouli et al. (2005b, p. 426) explicitly counts missing elements - when compared to an expert’s map - which negatively impact the total score of a map. Concerning the elements that *are* present in a concept map, they are typically treated as an artifact of a person’s knowledge structure. It is possible of course that this person has only guessed them and that they are not the result of a process of integration and thus not reflect anything from the knowledge of the person. Whether or not this has been the case is typically not observable in the outcome, though. However, nearly all methods of externalization are prone to this flaw, as the process itself is not easily observable.

The rest of this section investigate some of the (arguably) most influential factors on the externalization in a concept mapping task. Clearly, the motivation of the mappers is pivotal for the results. Also, the time for the concept mapping task and the training in concept mapping are heavily influential factors. In literature, there doesn’t seem to be a clear indication of how much time is needed for a given complexity in the maps. Some results concerning this problem are presented in the next section. It is also

unclear how much and what kind of training is necessary before concept maps can successfully be used as a method of externalization. Typically, the training found in literature is based on some explanations about concept mapping, an example map, and, often, a training map that persons are asked to produce before the “real” map. The aspects of motivation and training are especially important for studies consisting of several, repeated measurements of the same person. In this case, a development in the maps complexity, positive or negative, might just be the result of an increasing fluency in concept mapping or a decreasing motivation to keep drawing maps. In general, it will be very difficult to quantify the extent of influence that these variables have on the result of the externalization. So, it is probably best to judge on a case by case basis about whether or not it seems plausible to suspect a visible influence on research results, based on the actual data.

Since concept maps are relying on language it is also important to keep in mind the effect that expressing a mental idea in language has: “Concepts also take part in a bidirectional relationship with language. In particular, one’s repertoire of concepts may influence the types of word meanings that one learns, whereas the language that one speaks may influence the types of concepts that one forms” (Goldstone & Kersten 2003, p. 613). Concerning a list of concepts, Cooke (1994, p. 825) notes that “it must be kept in mind that the stimuli are *words*, not concepts, which may be interpreted differently by different individuals”. Going even further than that, Solomon et al. (1999, p. 100) state that “[c]ommunicating about an entity also appears to affect conceptual structure and categorization”. While this is usually taken as a good sign, in the sense that it fosters learning, it also succinctly expresses the inherent problems of externalizing knowledge: As has been argued in chapter 4, drawing a concept map is a form of communicating, fosters meta-cognition, and also is an act of knowledge creation. This, in turn, may affect the knowledge structure that is to be externalized. In other words the method of measurement may affect the very aspect to be measured. Consequently, Trumpower et al. (2010, p. 9f.) suggest that using relatedness judgments instead of concept mapping is a better way of arriving at structural information because it doesn’t require persons to label the links and may therefore be less dependent on language skills. Also, the explicit structure of a map, as a person is drawing it, can have a bias on the externalization process since it may foster the creation of “visually or structurally appealing” (Trumpower et al. 2010, p. 10) maps regardless of the actual knowledge structure that should be externalized. However, at least in the end of a concept map creation process, the changes in the knowledge structure that were triggered by concept mapping will most probably also have been integrated in the map. At this point the knowledge structure and the map are consistent again, then.

In conclusion, it is assumed that based on the vast amount of literature of successful concept mapping tasks and the psychological backings of the method, at least the “gold standard” task as described in section 4.2.2 with the common restrictions of a list of concepts or a restricting list of concepts is a valid approach to externalizing parts of the conceptual knowledge of a person. However, the externalization is influenced to an unknown extent by a set of variables not fully known and therefore all results must be treated with caution in analysis.

### 6.1.1 Observing the Externalization

This section presents several results concerning the process of concept map creation that have been gathered in the course of this thesis. They are concerned with three different aspects: The general complexity of concept maps achieved in assessment tasks, the actual process of creating them - especially regarding the required time - and the choice of labels. All three aspects are interesting, since they provide insights into the influence of variables on the process of knowledge externalization during concept mapping. There are two possible methods of investigating this process: Either the process is observed (unobtrusively), or the drawing persons themselves are asked about the externalization process, for example by interviewing them afterwards or by having them comment on the creation process. This itself is an externalization again. Both methods will have their respective strengths and weaknesses and one cannot completely replace the other. Yin et al. (2005) used the think-aloud technique to investigate the generation process. They found that supplying a list of pre-defined linking phrases seemingly changes the cognitive process underlying map creation. Concerning the value of observing the creation of a concept map, Cañas (2008, p. 62) notes: “Following the evolution of propositions over a given time span, furthermore, can help visualize the process of meaningful learning as revealed by subsumption, progressive differentiation and integrative reconciliation of concepts, link reworking, and overall map reorganization.”

As part of this thesis, an online editor for concept map creation has been designed, as described in chapter 8. This allows a completely unobtrusive monitoring of the process of map creation. The editor has been used in several different applications and contexts: Some of the maps were used as part of an exam at the university, others were drawn completely voluntarily as a learning aid, and some were collected as part of a survey. Also, in some of the cases, a list of concepts or a restricting list of concepts was used, while in others there were no restrictions whatsoever. Most of the maps were drawn by learners, some were done by experts for the respective setting, namely secondary school teachers and advanced students.



One general observation is, that when asking learners to draw concept maps, the result will typically be rather sparse, tree-like graphs (see section 6.2.1 below). Of several hundred concept maps that were collected in various contexts, including all of the maps used in the case studies of the fourth part, the number of concepts and propositions were counted. When correlating the number of concepts to the number of propositions using Pearson's product-moment correlation, with  $p < 0.01$  the value lies in the interval of 0.98 and 0.99. So, there is a clear linear dependence between the number of concepts and the number of propositions. Also, the mean value of the quotients of the number of propositions divided by the number of concepts over all maps is 0.98. So, it seems that not only is there a linear trend between the number of concepts and propositions, but the linear factor is close to 1, i.e. concept maps are mostly resembling trees. While it may seem that a densely connected conceptual knowledge is preferable, a very high proposition to concept ratio is not considered optimal elsewhere in literature (cf. Glögler 1997, p.136).

To gauge the effect of using a software-based approach, all of the maps that were not drawn using any form of software were analyzed in comparison to all the maps that were drawn using the editor. The average ratio of propositions and concepts for the non-software maps is 0.92. Also, a t-test performed on both sets of these ratios (number of propositions divided by number of concepts for every single map), reveals a significant difference ( $p = 0.014$ ) between the two populations. So, it seems that using a software-based approach influences the density of the maps in a small but positive way.

Next, the actual process of creating the maps is analyzed in more detail. The online editor records a snapshot of the concept map after each of the basic operations of adding, removing or renaming a concept or proposition. The simple moving of concepts on the plane is not recorded. The snapshots can be used to identify the working pattern used by the creators. One typical working pattern might be, for example, to always create a new concept and then a new connection using this concept. A different working pattern, and one that might more closely resemble an exhaustive externalization process, would be to create a few concepts and then creating many propositions between these concepts. To analyze the pattern, the difference between consecutive snapshots was identified and the operation that was used between the snapshots was marked with a code (e.g. 'C' for creating a concept). Then, the sequence of codes of all the snapshots of a map describes the way the concept map's structure has been created over time. For example, if the codes 'C' and 'P' stand for adding a concept or proposition respectively, then one pattern of creation might have the code "CPCPCPCPCP"; another pattern could be "CGPPCCGPPCPCPP". The results of all collected maps in the editor show

that the average length of consecutive operations of either adding only concepts or adding only propositions is between 1.5 and 2.5. In other words, in general, at most two concepts are added before a proposition is added or at most two propositions are added before a new concept is added. This clearly points to a working pattern of repeatedly only connecting new concepts to an existing map structure - which will result in a tree like structure.

A variable that will heavily influence the externalization process is time: If not given enough, the externalization of a person's knowledge in a concept map must necessarily fall short regarding validity. In literature, there doesn't seem to be an established amount of time that is necessary for a concept mapping task. The time that was spent between each action in the concept mapping editor is recorded and can be analyzed. In the course of two runs of the same lecture described in section 6.3.2, the students had to draw concept maps as part of their final grade but were not given any time restrictions (apart from a final due date), as the concept mapping task was done at home on their own schedule. When not differentiating between creating concepts and creating propositions, a histogram of the time spent between each action can be seen in Fig. 9. An average of about 20 seconds (17.51) is used for each action, the standard deviation is 14.56 seconds. Since the collected maps are also rather sparse with a concept to proposition ratio close to 1, as reported above, it can be assumed that if given a list of  $n$  concepts, a time of at least  $n$  minutes should be enough to not negatively influence the externalization - at least when the maps are drawn using software. Whether or not this also holds true for maps created with pen and paper is unclear, but at least the actual performance of typing and clicking versus writing can be assumed to be of roughly equal duration. However, time is only partly a constraint when considering the number of concepts given in a concept mapping test. For example, Clariana & Taricani (2010) report that the predictive ability of concept map scores decreased when increasing the number of concepts that students should use.

Finally, the labels that are used in concept mapping are investigated. Going even further than the categorization of different types of propositions presented in section 4.1, from a researcher's perspective, it would be favorable to have the mapper choose linking words from a given list of labels - since this would greatly ease automated analysis and scoring of large amounts of maps. However, such a list could also negatively influence the externalization. While Yin et al. (2005) found a difference in cognitive processes when supplying a list of labels, this list has been extracted from an expert's map. If, instead, a list could be found that was independent of the subject matter context and provided a reasonable amount of coverage over possible concept relations, a different outcome can be expected.

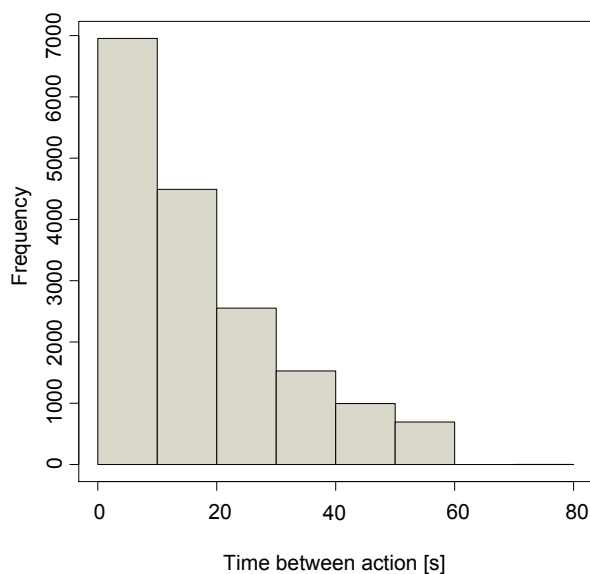


Fig. 9: The time between consecutive actions in the concept mapping editor for 58 (rather large) concept maps of students of two runs of the same lecture. The maps were drawn with no time restrictions and, in general, over several drawing sessions.

Cooke (1983) presents a classification of link types for similarity ratings concerning the semantic relation specifically between concepts of computer programming. The types are *can be done to/is operated on*, *is part of/contains*, *interchangeable/interchangeable*, *is a type of/superset*, *is a type of/could be*, and *can be done to/is used with* (cf. Cooke 1990, p. 238). The two wordings are used since the types can be used in both directions for a pair of concepts. Also, Sousa (2009, p. 200) gives the results of a study that identified nine types of cognitive relationships for concept mapping with the following names and descriptions:

1. *Classification*: A is a an example of B.
2. *Defining/subsuming*: A is a property of B.
3. *Equivalence*: A is identical to B.
4. *Similarity*: A is similar to B.
5. *Difference*: A is unlike B.
6. *Quantity*: A is greater/less than B.

7. *Time sequence*: A occurs before/after B.
8. *Causal*: A causes B.
9. *Enabling*: A enables/allows B.

Clearly, from an epistemological view as presented in the next section, these categories are not enough to encode arbitrary propositional facts, like “the earth rotates around the sun” which is clearly neither a causal nor an enabling relationship between the concepts *earth* and *sun* nor any of the others. The system of Cooke is not applicable due to the different context, of course. In the course of the case study presented in chapter 10 in the next part, the labels that students were actually using in a concept mapping task were analyzed. The results have been published in (Hubwieser & Mühlung 2011c). Students were free to choose any labels during creation. Later on, these labels were categorized by a qualitative approach. To this end, each label was rephrased by stripping unnecessary prepositions and converting it into an active present tense form, as far as possible. These were then categorized by combining them under some appropriate label. For example the labels “characterizes” and “describes” are combined into the category “describes”. The process is subjective to some extent. To increase objectivity, there was also a category “too complex” that contains all labels that could not easily be placed into a category described by a present tense verb. The categories of the labels that are not too complex and that were actively used by at least 3 students are only the following 29 (translated from German):

*activates, belongs to, calculates, changes, concretizes, connects, consists of, contains, controls, corresponds to, creates, describes, determines, enables, has the effect of, fulfills, gets, influences, is divided into, is entered, is opposite of, is rejected, works with, names, needs, repeats, results in, uses.*

Two only appeared in passive form and were therefore left that way. Many of the categories can be placed in the two systems presented above. For example, “enables” and “needs” belong to “Enabling” while “determines” and “results in” can be placed into the “Causal” category. Some of the resulting categories, like “calculate” or “activates” are probably a result of the specific context of computer science. In other contexts, like biology, other categories will most probably appear, while “calculate”, for example, probably won’t.

So, it seems like providing a list of linking words that is encompassing enough to allow for arbitrary concept mapping tasks is a difficult endeavor. Also, it remains unclear in how far it would influence the externalization process. Automated scoring of propositions would be enhanced by a given list of labels. Using computer support

for drawing concept maps allows a more flexible approach to this problem, as also done by Weber & Schumann (2000, p. 162): Instead of giving a fixed list of labels, a computer system might suggest several labels while still allowing users to create their own linking phrases if they like. In this way, they would not be restricted but guided toward reusing labels of other users.

## 6.2 Epistemological View

From an epistemological point of view, mainly the propositions encoded in concept maps are of interest. The same information can also be transported by, e.g. a list of the propositions as sentences. The structure of the concept map (e.g. hierarchical or taxonomy like) is then merely a chosen way of displaying the knowledge encoded in the propositions that makes the map easier to grasp visually, for example. From an epistemological perspective, a proposition should state some form of “fact” in the given subject matter of the concept map and not just encode trivial connections that can be made up between almost any pair of concepts like “has something to do with”.

If, as in the last section, concept mapping is assumed to be based on the mental process of integration, the fact that a proposition encompasses (exactly) two concepts is not a restriction. Novak, as presented in section 4.1, defines a proposition such that it connects two or more concepts and Larraza-Mendiluze & Garay-Vitoria (2013, Figure 8) show an example of a triadic proposition in a concept map, that is split up for better readability. However, it remains unclear how such propositions are compatible with the way concept maps work. While strictly speaking a proposition that connects a concept to itself (e.g. “objects are communicating with objects”) is not covered by the definition, it is no problem to integrate such an association in a concept map conceptually and visually. Concerning propositions with more than two concepts, there is no obvious solution, however. For example a simple propositional fact like: “The result of a transition (for a given deterministic automaton) is determined by its current state and the next input” is not easily transformed into a proposition between two concepts. If the context of a given deterministic automaton is clear, the concepts that are present in this sentence are: *transition*, *state* and *input*. Therefore, the proposition is not dyadic (based on two concepts) but triadic.

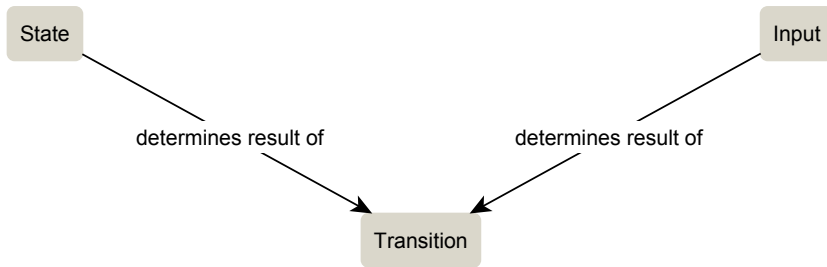
A concept map containing these three concepts with meaningful connections regardless of the given proposition can easily be created. However, the aforementioned propositional fact cannot be encoded in such a map in an unambiguous manner. Basically, there are two different ways of how it can be attempted without changing

the syntax of concept mapping. The triadic proposition can be split up into several dyadic ones, as in Fig. 10(a). Then, the propositions are formed correctly, but taken in isolation they are wrong. Instead, they would have to be interpreted as a set of propositions that are combined in some way. Without specifically hinting at this problem, Cañas (2008, p. 62) states that a proposition in a concept map is not valid if, among others, “it is not autonomous, i.e., it is a fragment or continuation of a larger grammatical structure such as a sentence, and has no meaning independently of this bigger structure”. Alternatively, the proposition can be encoded by using more “complex” linking words encompassing other concepts, as in Fig. 10(b). In this case, each proposition is correct by itself and, in this example, redundant. As a matter of fact, the concept map of Fig. 10(b) would contain the same information, if one of the concepts *input* or *state* was missing completely. However, it defies the semantics of concept maps that concepts are appearing as separate elements. Also, it defies the notion of a well-constructed concept map presented in section 4.1, according to which only short labels should be used for propositions. The same goes for using more complex concept labels instead of complex propositions. The first case study presented in chapter 10 will shed some more light on this restriction by analyzing different patterns of splitting up a proposition.

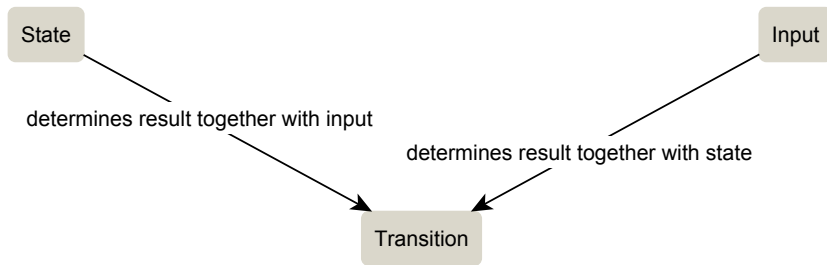
### 6.2.1 Concept Maps as Graphs

This section will establish a formal definition of several typical forms of concept maps, in order to provide a basis for the following definitions of concept landscapes. The formalism is kept minimal on purpose in order to make the definition usable and easy to communicate.

It is self-evident, that concept maps in their most general form can be modeled mathematically as little more than a labeled, directed graph, as noted in e.g. (Leake et al. 2005), (Koponen & Pehkonen 2010), or (Anohina-Naumeca 2012). The concepts form the nodes of the graph and the labeled arrows form the edges. Care must be taken, however, that a concept should appear only once in a concept map. So, the basic model of a concept map  $CM = (V, E, L_v, L_e)$  consists of a finite (and usually non-empty) set of vertices and edges that form a directed graph and two functions  $L_v : V \rightarrow \Sigma^*$ ,  $L_e : E \rightarrow \Sigma^*$  that map the set of vertices and edges to the set of all words  $\Sigma^*$  formed over some alphabet  $\Sigma$ . Additionally, it must hold that  $L_v$  is injective, i.e. that  $L_v(x) = L_v(y)$  if and only if  $x = y$ . A proposition, then, for some edge  $p = (x, y)$  is the triple of the labels:  $(L_v(x), L_e(p), L_v(y))$ . The usual application of concept maps assumes that this triple, if read as a “sentence”, will be understandable for human readers. The labels can be defined more precisely (e.g.



(a) Splitting the proposition up



(b) Using complex linking words

Fig. 10: A triadic proposition encoded in a concept map.

by formal languages), to restrict concepts to a single word, for example. Also, though not very common, self-loops are allowed in this definition and also sometimes found in real-world concept maps, as already noted above.

A concept map that is restricted to a list of concepts and/or to a list of edge labels can be modeled as follows - both restrictions can be applied simultaneously:

**Concepts** Let  $C \in \Sigma^*$  be a set of concepts (given as labels). Set  $V_C$  to  $\{v_1, v_2, \dots, v_{|C|}\}$  and define  $L_c : V_c \rightarrow C$  as a bijective mapping function. Then, the concept map  $CM_C = (V_C, E, L_C, L_E)$  for some sets of edges and labels is restricted to the concepts of  $C$ .

**Edges** Let  $L \in \Sigma^*$  be a set of edge labels. Define  $L_E : E \rightarrow L$  as a mapping function. Then the concept map  $CM_E = (V, E, L_V, L_E)$  is restricted to the edge labels of  $L$ .

Given a set of concepts  $C$ , it is sometimes convenient for analysis to restrict an existing map to the concepts of  $C$ . This is equivalent to using the subgraph induced by the vertex set  $L_V^{-1}(C)$ . This uses the inverse function of  $L_V$  applied to a set of labels.

Adding the restriction that a concept map should be hierarchical is not straightforward. A hierarchical graph or network in graph theory is typically seen as one that “divides naturally into groups and these groups themselves divide into subgroups, and so on until we reach the level of individual vertices” (Clauset, Moore & Newman 2007, p. 2). A hierarchical concept map on the other hand is one where the concepts can be divided into several levels, starting from one central concept. One level subsumes the nodes of the next level, structurally and semantically, “that is, the more general, more inclusive concepts should be at the top of the map, with progressively more specific, less inclusive concepts arranged below them” (Novak & Gowin 1984, p. 15f.).

While this bears some resemblance with the graph theoretic definition of a forest (a graph containing no cycles), a hierarchical concept map can also contain edges within one level of the hierarchy and also edges that are not strictly between adjacent levels (cross-links) - which may potentially form cycles in the graph. One solution is to define a hierarchical concept map as a regular concept map with the restriction that there must be a partition of the set of nodes, such that:

1. One subset of the partition contains exactly one node. This node is the “root”, i.e. it forms the topmost level of the hierarchy /concept map.



2. There must be an ordering of the subsets of the partition, such that “most” edges that are not connecting nodes within a subset are connecting nodes of adjacent subsets according to the ordering. The edges that are connecting nodes of non-adjacent subsets are cross-links.

Then, each subset forms a level of the hierarchy. For ways of determining the root node of a concept map, see e.g. (Valerio et al. 2008). Also, this definition of cross-links only holds for their original meaning. The different meaning of a cross-link connecting different segments of a map can be formalized using graph communities as presented in the section 7.2.3.3.

In the analysis of concept maps, it is often required, that the propositions of a map are scored. This score can then be used for a more detailed evaluation. A scored concept map  $CM(V, E, w, L_V, L_E)$  is a weighted graph, where the weight function  $w(i)$  denotes the score of the proposition formed by edge  $i$  and its incident nodes.

Also, for concept landscapes presented in the next chapter, a graph can have a layout and a temporal development (i.e. development over time). The layout can be modeled by a function  $P : V \rightarrow \mathbb{R} \times \mathbb{R}$  that places each vertex on the two dimensional Euclidean plane. While the concept of time is a very complex subject by itself, dynamic sets only need a very limited definition of time. There are several discrete and distinct points in time (without any duration)  $t_1, t_2, \dots, t_n$  and a (total) temporal ordering of these points. In other words, for each pair  $t_i, t_j$ , it must be possible to determine whether  $t_i$  was before  $t_j$  or vice versa. This is a simple notion of discrete time events. In literature, there are several models of such temporal or dynamic graphs to be found. Commonly, it is either modeled as a series of graphs (cf. Holme 2003) or as a series of operations (e.g. adding an edge) on a graph (cf. Mehta 2005). Both methods are applicable for concept maps as well. While the analysis presented above concerning the sequence of operations is more influenced by the second way, the analysis of concept landscapes is often more akin to the first way of modeling. A possible way of formalization is based on dynamic sets of vertices, edges and labels, denoted as  $V(i), E(i), L_V(i)$  and  $L_E(i)$  respectively.  $i$  denotes a point in time and the dynamic set may be different, i.e. contain different elements, for each point in time.

### 6.2.1.1 Representations in Computers

When working with concept maps in computer-based analysis, it is necessary to represent the concept map digitally in a suitable format. The most common representations of graphs in a computer are (cf. Cormen et al. 2001, p. 527ff.):

**Adjacency Matrix** For a set of  $n$  vertices, a  $n \times n$  binary matrix is constructed, where an entry  $a_{ij} = 1$  if and only if the nodes represented by the  $i$ -th row and  $j$ -th column are connected (adjacent) in the graph. For undirected graphs, this matrix is symmetrical.

**Adjacency List** For a set of  $n$  vertices, an ordered set containing  $n$  lists is created, where the  $i$ -th list contains all adjacent nodes of the  $i$ -th node.

**Incidence Matrix** For a set of  $n$  vertices and  $m$  edges, a  $m \times n$  binary matrix is constructed, where an entry  $a_{ij} = 1$  if and only if the edge represented by row  $i$  is incident to the vertex represented by column  $j$ .

**Incidence List (Edge List)** For a set of  $m$  edges, a list with  $m$  entries, where the  $i$ -th entry contains the two nodes that are incident to the edge corresponding to that entry. It can also be represented as a  $m \times 2$  matrix, where each row corresponds to an edge and contains the two incident nodes of that edge in the two columns.

For concept maps, it is sometimes convenient to include the edge labels in the representation (e.g. by not storing an edge list, but instead a “proposition list”, or by using the edge labels as entries in the adjacency matrix itself). When the chosen method of analysis ignores the edge labels, it is usually more convenient to treat the concept map as an undirected graph. The direction of an edge is basically only dependent on the chosen wording of the proposition and usually an equivalent alternative wording can be found that reverses the direction of the edge. In this case, an undirected graph contains the same structural information, as also noted by Anohina-Naumeca (2012).

In addition to these representations, this work will make use of the following two representations of concept maps, even though information is lost in the transformation:

**Concept Vector** For a given (ordered) set of concepts  $C$ , a binary vector  $v \in \mathbb{B}^{|C|}$  is created, where  $v_i = 1$  if and only if the  $i$ -th concept of  $C$  has at least one incident edge in the concept map.

**Edge Vector** For a given (ordered) set of concepts  $C$  and the (ordered) set of edges of the complete graph using  $C$  as set of nodes  $E_C$ , a binary vector  $v \in \mathbb{B}^{|E_C|}$  is created where  $v_i = 1$  if and only if the two concepts that are connected by the  $i$ -th edge of  $E_C$  are connected in the concept map.

### 6.3 Educational View

“Consistent with constructivist epistemology and cognitive psychology, the theoretical framework that supports the use of concept mapping hinges on the notion of meaning. Because meaning is both constructed and shared, the expectation of the student as an active agent in participating in the development of understanding is implicit.” (Edmondson 2005, p. 36)

The “educational view” on concept mapping is, in a way, a reconciliation of the previous rather extreme views. It is - in this work - based on the notion, that learners are creating concept maps as a form of assessment or measurement of their structural knowledge. This requires that concept mapping is both valid and reliable and that its limits and potentials regarding assessment are clear. From the previous two perspectives, the validity concerning both the externalization of structural information of conceptual knowledge and the encoding of knowledge in the form of (dyadic) propositions has been established. Using the “gold standard” of concept mapping requires that persons draw “real” concept maps, in the sense that they must explicitly give useful edge labels. Even if the labels are ignored in the subsequent analysis of the maps, this ensures that the properties of the structural configuration are not overestimated. A connection between concepts should only be given if there is an associated “element” of (conceptual) knowledge that the mapper retrieved from memory by using the mental process of integration or (at least) the recall of some factual information.

Concept mapping from an educational perspective also often entails an educational process that is assumed to influence the knowledge structure. In other words, concept mapping may be used to “monitor” how a learning opportunity affects a person’s memory. As in most forms of assessment however, the result of a concept mapping task is first and foremost an attribute of the person assessed and not of the educational process. Implicitly, often it will be assumed that it is this process that principally determines the result of the assessment. For example, when students are taking a math test in school, the results of the test are (hopefully) measuring the students’ knowledge or abilities in math. These may or may not have been a result of the math classes they attended. Implicitly, though, the assumption will often be made that it indeed is a direct result of the attendance though obviously combined with personal attributes and efforts of the students (e.g. preparing for the test).

In this work, it is assumed that the most gain from analyzing concept maps comes from the *structural information* of the maps and not from the list of propositions that

they encode. This structure is present both on a macro level, encompassing all concepts, and on a micro level, encompassing only a small number of concepts, and their interrelations. There are also other possibilities, of course, of arriving at such an externalization. For example the structural information could be externalized by relatedness judgments and the propositional information could be collected afterwards (cf. Albert & Steiner 2005). Also, concerning the epistemological limitations, when concept mapping is mostly seen as a way of organizing and displaying knowledge, there are also better alternatives, due to the restrictions of concept maps. However, concept mapping has the inherent advantage that it is easy to learn for most people - in contrast to e.g. describing a knowledge base as an ontology - and it offers a direct benefit for the mapping persons - in contrast to e.g. relatedness judgments, especially when the mappers can keep or continue working on their concept maps. Also, concept maps are a valuable tool of formative assessment, in contrast to many other methods of externalizing knowledge. Formative assessments encompass “all those activities undertaken by teachers, and/or by their students, which provide information to be used as feedback to modify the teaching and learning activities in which they are engaged” (Black & Wiliam 1998, p. 7). This makes concept maps valuable for applications in educational settings as there is more to be gained than “just” the externalized knowledge.

### 6.3.1 Assessing Learning

Concept maps, when used as a form of assessment, may be used to measure whether or not an intended learning objective (see section 3.3.1) has been met by students. This section deals with the question of which types of learning objectives a concept mapping task can cover, in principle. It is based on the taxonomies described in section 3.3.1.1. Cañas & Novak (2012, p. 248) note without further elaboration that “when concept maps are used to facilitate learning, they can also be used as an assessment tool capable of assessing not only the recall of information but also those higher order skills that are described in Bloom taxonomies”. And, according to Novak & Cañas (2008, p. 13), the process of identifying cross-links involves “high levels of cognitive performance, namely evaluation and synthesis of knowledge”. Again, the authors don’t give a reason for this classification.

The prototypical learning objective that can be assessed with concept mapping, following the taxonomy of Anderson and Krathwohl, is “understanding conceptual knowledge”. This is an oversimplification, though. As has been argued above, drawing a proposition can be a simple task of recalling a fact (“remembering factual knowledge”) or it can require understanding a more complex interplay of several

knowledge elements, depending on whether or not the proposition (e.g. “parameters can be passed by value”) has been rote-learned as a “fact” by the person drawing the map or not. Clearly, by design, concept maps will fall short for assessing learning objectives in the realm of procedural knowledge. Similarly, the cognitive process “apply” is typically not easily covered by a concept mapping task. This combination is not surprising: “*Apply* involves using procedures to perform exercises or solve problems. Thus, *Apply* is closely linked with *Procedural knowledge*” (Anderson & Krathwohl 2001, p. 77). However, when including aspects like the layout of the concept map, producing a “good” layout can be considered an application of meta-cognitive knowledge.

To investigate whether or not the higher-order cognitive processes can be reached by concept mapping, the subtasks that are needed to draw a concept map are analyzed, as if they were expressed as actual learning outcomes for some (hypothetical) subject matter. For example the subtask of “identifying key concepts” that is analyzed in the following actually stands for an operationalized learning outcome that could be, for example, “the student is able to identify key concepts of object orientation”. The knowledge component is taken to be either factual or conceptual knowledge. The task of creating a concept map is defined by Novak & Cañas (2008) as:

“Given a selected domain and a defined question or problem in this domain, the next step is to identify the key concepts that apply to this domain. Usually 15 to 25 concepts will suffice. These concepts could be listed, and then from this list a rank ordered list should be established from the most general, most inclusive concept, for this particular problem or situation at the top of the list, to the most specific, least general concept at the bottom of the list.[...] The next step is to construct a preliminary concept map.[...] Once the preliminary map is built, cross-links should be sought. [...] Finally, the map should be revised, concepts re-positioned in ways that lead to clarity and better over-all structure, and a ‘final’ map prepared.” (Novak & Cañas 2008, pp. 11-14)

Anderson & Krathwohl (2001, Table 5.1) give detailed descriptions about the cognitive processes of the taxonomy. Applying these to the task description above, the following matches can be identified by interpreting the description of the concept mapping task as described and looking for correspondences to the descriptions of the taxonomy - especially by using the list of synonyms that is given for each of the cognitive processes. Also, Sousa (2009, p. 250ff.) presents a description of the different cognitive processes that is used for the interpretation. The analysis yields:

- **“identify the key concepts”** While “Identifying” is given as an alternative name of the sub-category “Remember-Recalling”, in this case it means “selecting” key concepts. “Select” is given as an alternative name for the sub-category “Analyze-Differentiating”. Also “identifying parts” is given as a description of the process “Analyze”.
- **“from this list a rank ordered list should be established”** Ordering is a form of structuring, which belongs to the sub-category “Analyze-Organizing”, also “recognizing organizational principles” is given as a description of the process “Analyze”.
- **“construct a preliminary concept map”** Constructing a concept map is rather general. Particularly, it involves arranging the selected concepts and deciding which to link and how to name the links. Therefore, the task encompasses retrieving relevant information about the concepts from memory (category “Remember-Recalling”), selecting propositions to be included in the map (category “Analyze-Differentiating”) and integrating those propositions into the map (category “Analyzing-Organizing”). “Recall of semantic memory” is a description of the category “Remember” and “examining the relationships of the parts to each other and to the whole” is given as a description of “Analyze”.
- **“cross-links should be sought”** Again, this subtask is too general to be directly analyzed. It encompasses tasks of retrieving information, selecting and integrating, and therefore fits the cognitive process of “Analyzing”. For example, “organize and reorganize information into categories” is given as a description of “Analyze” which can be applicable for finding cross-links. As there are no “criteria and standards” that drive an evaluation process, choosing “Evaluating” as the cognitive process is not applicable in this case.
- **“the map should be revised, concepts re-positioned in ways that lend to clarity and better over-all structure”** Assuming that a concept map should be hierarchical, this can be seen as a criterion. Then this subtask may be seen as “Evaluate-Critiquing”. Also, “learners tend to consolidate their thinking” is given as a description for “Evaluate”. Otherwise, it is more a form of structuring and classification, as above, corresponding to sub-categories of “Analyze-Organizing” and “Understand-Classifying” respectively.

In conclusion, a concept mapping task encompasses several subtasks that are placed differently in the taxonomy. The “highest” category that a regular concept

mapping task according to this analysis can “safely” reach is “analyzing conceptual knowledge”. When using the taxonomy of Fuller et al. (2007), then a concept mapping task can cover almost every aspect of the “Interpreting” dimension - which makes concept mapping especially interesting in the context of computer science which this taxonomy is specifically based on.

In a related investigation, Passmore (1999) uses the SOLO taxonomy of observed learning outcomes on elements of concept maps. Propositions are assessed according to the taxonomy and this in turn is used to judge the quality of a concept map. According to his reasoning, propositions can be of any type up to “extended abstract”. However, this is only possible, because the propositions are extended with actual explanations of the underlying reasoning and concept maps are used to summarize an instructional text instead of purely externalizing knowledge:

“Method 1 [out of three Methods of assessing concept maps] is an adaption of Biggs and Collis’ SOLO knowledge assessment method. It rates concept map links against the SOLO response scale. Links consists of one or more items of data and a linking statement. Each item of data expresses one element of the relationship between two concepts. [...] The understanding contained in this connection is expressed in the linking statement. Method 1 determines the quality of this understanding.” (Passmore 1999, p. 71)

It is questionable whether a simple proposition of a typical concept map can be accurately assessed using the SOLO taxonomy. However, it seems plausible that an entire map, especially when contrasted with some form of instructional input, can be assessed this way.

Clearly, concept maps are not a suitable tool to measure competencies - except for the competence of concept mapping itself. However, as has been presented in section 3.3.2.2, a competence usually entails a cognitive (or knowledge) component that is a prerequisite of competent performance. As has been established in this chapter, concept maps can be used to assess this cognitive aspect of a competence - at least as far as it concerns conceptual or factual knowledge. Therefore concept maps are deemed to be a suitable instrument of measurement even in the light of competence based assessment scenarios.

$\rho$	Concept Map	SUM	PT	EXP	WIKI
Concept Map	-	0.50*	(0.17)	(0.04)	(-0.04)
SUM		-	(-0.07)	(0.19)	(0.20)
PT			-	0.87**	(0.12)
EXP				-	(0.09)
WIKI					-

Table 6.1: Correlation between different elements of the assessment. Spearman's rank correlation was used (\*\* =  $p < 0.01$ , \* =  $p < 0.05$ , () =  $p > 0.05$ ).

### 6.3.2 Scoring Concept Maps

Scoring concept maps has a long tradition both in assessment and research. Often, the underlying assumption is that not all propositions should be treated equally. A proposition may be in contradiction to the views of experts on a subject matter and therefore indicate an inappropriate proposition or misconception. Also, a given proposition may be valued as "better" (mostly in the sense of "less trivial") than another proposition. Therefore, it often seems reasonable to score the propositions and subsequently use that scoring in the process of assessment (or analysis). However, as has also been mentioned in section 4.2.2, a scoring often introduces a subjective element and the reliability of the scoring scheme is highly important.

If concept maps are used as a form of formative assessment the scoring will have to be done in a way that allows insights into a person's particular strengths and weaknesses concerning the conceptual knowledge expressed in a concept map. This often entails scoring each proposition. In the case studies of the next part, a scoring using three values ranging from "clearly wrong or uninterpretable" to "clearly correct" and a middle value of cases where both extreme values are not applicable has been adopted. In analyses, the propositions with this middle value can be left out in order to further increase the reliability of the scoring. The scoring method is inspired by the "relational with master map" method presented in section 4.2.2. In the case study presented in chapter 10, the correlation of the summed scores of concept maps using this scheme to other structural attributes of the concept maps is investigated. The results which are presented in Table 10.1 show that there is indeed a medium to large effect present for the number of concepts, propositions, or components.

So, it seems that scoring each proposition in this way may only reveal information that is - in general - also present in other attributes of the maps. Since scoring each proposition of a set of maps takes a considerable amount of time, possibilities of



reducing the necessary manual efforts are worth investigating. When the scoring is done for analysis or grading only and not for feedback to the learner, this becomes especially interesting. One such method is presented here. The lecture that formed the basis for the first case study was completely restructured in the winter term of 2011 according to constructivist principles, as described in more detail in German by Berges, Mühling, Hubwieser & Steuer (2013). Instead of an exam, the students had to create a portfolio that consisted of several elements as described below - including five concept maps for the five major topics of the lecture. The concept maps were scored using the following very basic scoring scheme:

1. Count the number of concepts that are used ( $c$ ).
2. Evaluate the overall quality of the map holistically and assign a value of 0, 0.5 or 1 ( $q$ ).
3. Calculate the score as:  $\min(\lfloor \frac{c}{10} \rfloor \cdot q, 10)$ .

The maximal score of 10 points was set according to the maximal number of points that were awarded to the concept maps. The division by 10 was used for simplicity and because it fits the actual number of concepts well (in the sense that the results are spread out across the spectrum of 0 to 10 points). The holistic scoring was derived by checking whether or not propositions are actually labeled in a useful way and whether or not the concepts are making sense in the context of the lecture. So, the scoring scheme does not take the correctness of the propositions into account, based on the notion that misconceptions as expressed by the students may be valuable and should not negatively impact the score. The scoring system is not completely automatic, but the second step can be done very quickly, manually. This is akin to the manual classification of concept maps into the types of “spoke”, “chain”, or “net” which has been suggested to be a fast and reliable alternative to manual scoring of propositions (see section 4.2.2). The method has shown to be reliable between two different coders. To investigate the validity of the scoring system, the resulting scores of the concept maps were correlated with several other artifacts that were used for determining the grades, namely:

**SUM** The students were asked to write a summary of the topics of the lecture (at least 1000 words).

**PT** The students had to hand in several programming tasks over the course of the lectures. These could be done in groups, though.

**EXP** At the end of the semester, each student had to take part in an interview, explaining the solutions to the programming task individually.

**WIKI** As a creative element of the assessment, the students could write entries for a Wiki concerning the topics of the lecture.

Spearman's rank correlation of each of these elements with the score of the concept maps can be seen in Table 6.1. The strong (and significant) correlation of the concept map scores with the summary, is the second best value over all correlations - only bettered by the correlation of PT and EXP. A high correlation between concept mapping in written text has been identified before by e.g. Zimmaro et al. (1999). Also, seemingly, the different parts of the assessment are somewhat independent, with the programming tasks and explanations only weakly correlating with the rest. Overall, this indicates that actual programming tasks only partly capture conceptual knowledge. The case study presented in chapter 12 is investigating this phenomenon more closely. In summary, the results presented here and in the case study show, that a combination of qualitative and quantitative scoring with only few different score values and clear distinctions between quality levels works well and correlates with other methods that assess knowledge and abilities.

## 7 | Concept Landscapes

Based on the constructivist theory of learning and the cognitive and neurological backings of this theory as presented in chapter 3, it must be acknowledged that learning is a subjective, personal process and that teaching is far more a fostering of learning than it is a transportation of knowledge. Nevertheless, the knowledge organization of experts tends to be similar and when teaching in larger groups, the effects of teaching will most probably tend to affect the learners in a similar way. Investigating knowledge in a constructivist setting therefore can focus on two aspects:

1. The subjectively constructed, idiosyncratic knowledge structure of a person.
2. The commonalities and differences between the knowledge structures of several persons.

Concept maps have been established as a method of investigating the first aspect, namely the subjectively constructed knowledge structure of a person. Based on the findings of the last chapter and the related literature presented in the last part, it is clear that concept maps, when interpreted with caution, are a valuable tool in externalizing structural knowledge.

In this thesis and particularly in this chapter a generalization is presented, that allows using concept mapping for the second aspect, i.e. to investigate the “knowledge structure” of groups of persons in several distinctly different ways. One specific way of analyzing shared aspects of concept maps, which is also encompassed by this generalization, is the creation and subsequent inspection of a weighted graph that is formed from a set of maps. This approach is typical according to Eckert (2000, p. 5), for investigating shared knowledge structures and has been recently done by e.g. Larraza-Mendiluze & Garay-Vitoria (2013) who are investigating a set of concept maps with techniques of social network analysis. Also, Glöggler (1997) presents a study that investigates the development of knowledge by using a technique similar to concept maps and using the data to create a weighted graph that is then analyzed.

The specific setting requires new ways of working with concept map data but also presents new opportunities for insights in the data that are not possible when focusing on single concept maps. For example, as has been argued above, it is not valid to interpret missing elements for single maps. For groups this is different, as a hypothetical example in an instructional setting shows: If a single person is missing a

particular concept or connection, then this may be for a variety of reasons unrelated to the learning or the instruction this person received. If, however, every person of, for example, a class misses a particular concept or propositions, it becomes far more likely that this is due to their specific shared learning environment or input they received.

As has been mentioned in the last chapter, the externalization process during concept mapping is influenced by several variables. This influence can also be seen as “noise” that affects the measurement “concept mapping”. That the measurement of knowledge is inherently “noisy” has been observed at least for expert knowledge: “[A]lthough different experts may show variability in their judgments of concept relations, this variability often appears to be the result of random error rather than systematic differences in thinking” (Trumpower et al. 2010, p. 8). While the extent of “noise” in concept mapping in general is not known, it is at least plausible, that for many of the influencing variables, the “noise” over many measurements will rather cancel out than systematically add up. This is especially true for personal variables that influence the externalization. For example, when a group of persons is creating concept maps, it is reasonable to assume that the motivation will vary from persons with only little motivation for the task to persons with rather high motivation. Overall, therefore, the influence of a lack of motivation for concept mapping that is detrimental when assessing only a single, unmotivated person is far less severe for a group where only a few persons are unmotivated. The same can be expected to hold, within reasonable bounds, for the training necessary to create concept maps and the time given to create the map. Therefore, it is expected that the combination of many concept maps for analysis effectively reduces the influence of the main variables that may impact the externalization in a negative way.

The next section presents concept landscapes, which encompass the knowledge structures of a group of persons expressed in concept maps. Then, various analysis methods for concept landscapes are presented. These are applied in the case studies presented in the next part.

## 7.1 Definition

A *concept landscape* is a general notion for *aggregating* (or combining) the data of multiple concept maps with the goal of analyzing this combination instead of the single maps. The novel aspect therefore is to no longer treat a concept map as a single entity which is analyzed for itself, but instead focusing on analyzing sets of concept maps as a whole.

While in theory, any set of maps can be used to form a concept landscape, in practical settings the maps are typically taken from either of two scenarios: The first is the combination of maps that have been created by several persons in the same context, e.g. all students of a class at the same point in time. The second is the combination of maps that have been created by the same person but at different points in time. Fig. 11 shows these two scenarios as a diagram. It assumes, that a group of persons were asked to produce concept maps at several points in time. Then, allowing for some missing maps, the data will consist of a number of concept maps, where each map is associated with a person and a point in time. A *vertical* combination will be made up of all maps at a given point in time. A *horizontal* combination will combine the maps of each point in time, for a single person. More generally:

**Vertical** Combines a set of concept maps without a temporal ordering of the maps.

**Horizontal** Combines a set of concept maps with a temporal ordering of the maps.

Data mining approaches for example, are useful with vertical aggregations. Horizontal aggregations can be used for visualization of the developments of knowledge structures. Besides the choice of *which* maps to combine into a landscape, there are also two different ways of *how* this combination can be done.

**Amalgamation** The individual maps are combined into a single graph that is then analyzed. Often, this newly formed graph will be weighted in order to reflect some of the statistical properties of the original set of concept maps.

**Accumulation** The maps are still treated as individual entities, however information contained in them is combined in some form for analysis. Often, the maps will be transformed into a numeric vector and a matrix will then be formed from all the vectors.

Fig. 12 displays the difference graphically. An accumulation still treats each concept map as an entity that is recognizable in the data. For example, when forming a data matrix, each row will correspond to a specific map. When amalgamating, the single maps are integrated into a whole and the individual maps can no longer be identified. Usually both methods will contain loss of information: It is not possible to reconstruct the single maps from the aggregated data. Since the result of an amalgamation is a graph (void of edge labels and typically weighted) which is (almost) a concept map again, this can be used to combine several approaches: For example, the students of a lecture can be asked to create concepts maps at several points in time. Each

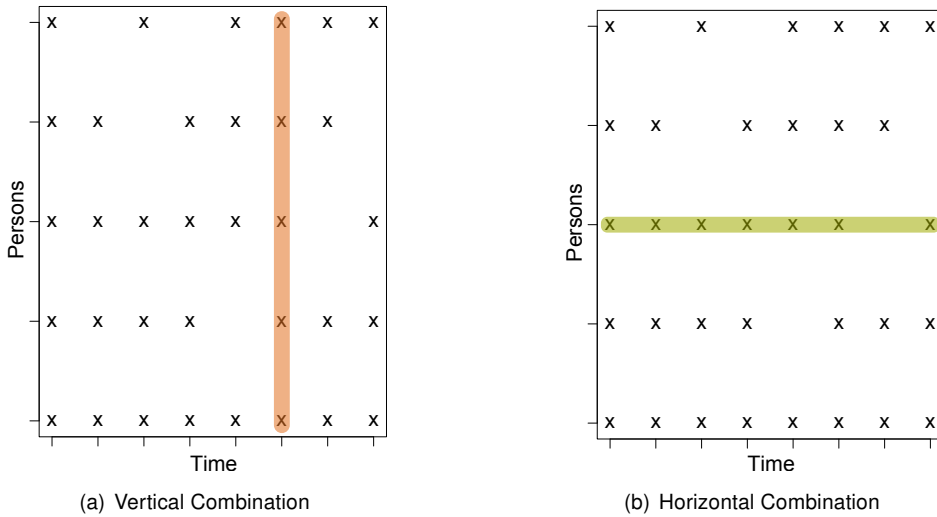
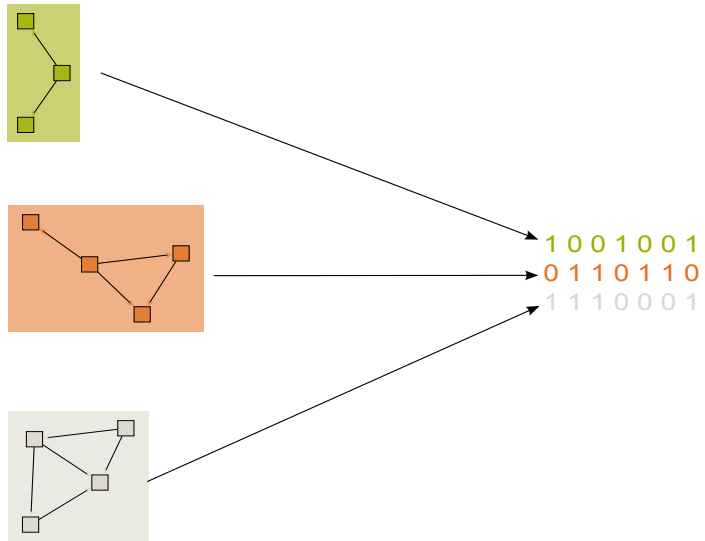


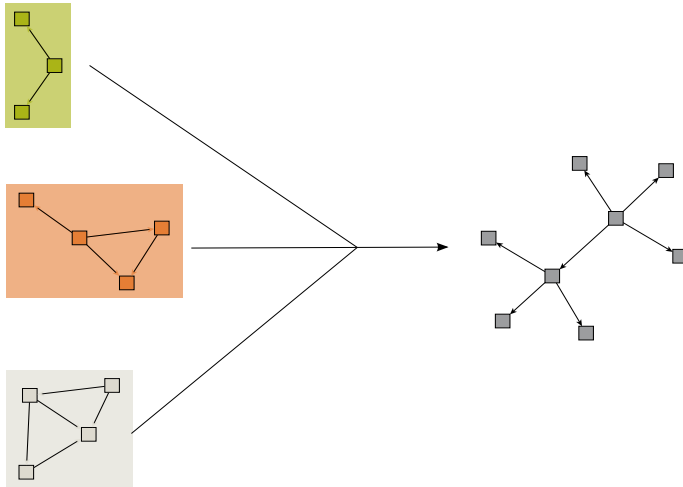
Fig. 11: A visualization of the two types of aggregations. Each 'x' marks the concept map of one person at one point in time. Some are missing. The aggregations either combine all maps of one point of measurement or all maps of one person.

point in time offers a set of maps that can be amalgamated vertically and then, the resulting graphs can be aggregated in either way horizontally in order to display the development of the groups' knowledge. This is schematically shown in Fig. 13 and applied in practice in the first case study. Other combinations may be used as well, of course. For example, a typical approach would be to first identify clusters of a vertical accumulation and then, in order to describe the clusters more closely, combine the maps that form each cluster again in a new way and search for the "common" elements in each cluster. This is possible because an accumulation still allows the identification of the constituent maps.

For vertical combinations, the focus of interest usually will determine which method is used: An amalgamation should be chosen, when the analysis focuses on the common elements of a group of maps. As the single entities are combined into a whole, the common elements typically will show more distinctly. When the focus of the analysis is placed more on the differences between the constituent maps of a concept landscape, an accumulation should be used. The single maps are still recognizable entities in this case and suitable analysis methods can identify how the single maps differ in the information contained in them. The two analysis methods that are described here and used in the case studies are using these two approaches: Pathfinder networks are used in identifying the "common" structural



(a) Accumulation



(b) Amalgamation

Fig. 12: Accumulations still treat each map as an identifiable entity in the resulting data. Amalgamations result in a graph or concept map. In both cases, the newly formed data is the input for the subsequent analysis steps.

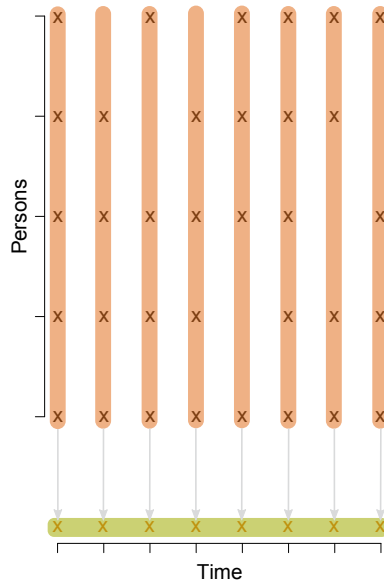


Fig. 13: A typical combination of aggregations: First, the maps of several points of measurement are amalgamated vertically, the resulting, new set of maps is then aggregated horizontally, using the times of measurements for the temporal ordering.

elements of an amalgamation of concept maps. For accumulations, cluster analysis is used to identify groups in the data.

By taking the type of landscape and the method of combining the maps into account, there are four distinct ways of how a concept landscape can be formed. By the nomenclature of typical research designs, a vertical accumulation corresponds to a *cross-sectional*-, or “*one shot cross-sectional*” study (cf. Seidel & Prenzel 2008, p. 282), in which a group of persons is studied at one point in time. A horizontal accumulation is a *longitudinal study*: “In a longitudinal evaluation, one or more clearly defined outcome measures are assessed repeatedly on a sample of subjects (e.g., students) over a period of time” (Ma 2010, p. 757).

For each of the four ways, a possible technique of working with the landscape in analysis is presented below. The next section will give a formal definition of the four types, built upon the formal definition of concept maps given previously.



### 7.1.1 Formal Definition

The formal definitions are based on the general notion of concept landscapes but, more specifically, on the analysis methods that are used in this work. Different methods may need a different formalism, but care is taken to keep the approach as general as possible.

For each case, it is assumed that there are  $k$  concept maps (as defined in section 6.2.1)  $CM_1 = (V_1, E_1, w_1, L_{V_1}, L_{E_1}), CM_2 = (V_2, E_2, w_2, L_{V_2}, L_{E_2}), \dots, CM_k = (V_k, E_k, w_k, L_{V_k}, L_{E_k})$  that are used for the landscape. Since the maps may be scored for analysis purposes, scored maps are assumed. For concept maps that are not scored, some constant weighting function like  $w(i) = 0$  can be used.

#### 7.1.1.1 Vertical

Vertical landscapes are simply a set of concept maps, combined into a new structure. A vertical amalgamation can be modeled as a weighted graph  $CL = (V, E, w, L_V, L_E)$ , where  $V = \bigcup_{i=1}^k V_i$ ,  $E = \bigcup_{i=1}^k E_i$ ,  $L_V(x) = \bigcup_{i=1}^k L_{V_i}(x)$  and  $L_E(x) = \bigcup_{i=1}^k L_{E_i}(x)$ . Based on the above definition, a single graph is created from the  $k$  concept maps. The definition of label functions ensures that no labels are “lost”. However, typically edge labels are not useful in the analysis of amalgamations. For concept labels, the definition again ensures no loss of information. It is most useful, when amalgamating a set of concept maps that share the same nodes and labels, i.e.  $V_1 = V_2 = \dots = V_k$  and  $L_{V_1}(x) = L_{V_2}(x) = \dots = L_{V_k}(x)$ . In this case, the set of concepts and their respective labels are staying the same for the landscape as well.

It is convenient to create a weighted graph and store some information about the constituent maps in the edge weights. Which information is used depends on the desired results and intended method of analysis, of course. For example, the information can consist of summing all the maps that connect a given pair of concepts and then using that number for the connection of this pair. Alternatively, instead of forming the simple sum, some additional filtering or transformation may be applied. For example, if the maps are scored, only maps with a score higher than some threshold could be used. Or only edges that are used in more than a given number of maps might be regarded in the analysis. The simplest form of only counting edges is defined as:  $w(e_i) = |\{1 \leq j \leq k | e_i \in E_j\}|$ , for  $w : E \rightarrow \mathbb{N}$ . A filtering that regards only edges with a score higher than  $t$  is defined as  $w(e_i) = |\{1 \leq j \leq k | e_i \in E_j \wedge w_j(e_i) > t\}|$ . Finally, a filtering that regards only edges that appear in more than  $t$  maps is defined as  $w(e_i) = \max\{|\{1 \leq j \leq k | e_i \in E_j\}| - t, 0\}$ .

For vertical accumulations, each constituent map is still an identifiable part of the “whole”. While there are many different possibilities, in the course of this work it is convenient (and sufficient) to model the landscape as a numerical matrix with  $k$  rows, where row  $i$  is a vector that is chosen to represent map  $CM_i$ . The concept and edge vector of a concept map as defined in section 6.2.1 are two possibilities for this representation. In this case, the resulting matrix will be called *vector matrix* and *edge matrix* for brevity. If the maps of the landscape do not share a common set of nodes, the definition of edge and concept vector have to be adapted to use the set of nodes or edges of the landscape instead of the set of the concept map. A more general model can be based on a mapping function that assigns each map a vector representation. So, a vertical accumulation is a matrix  $CL \in \mathbb{R}^{k \times j}$ , such that the  $i$ -th row is given by the value  $f(CM_i, \theta)$  of a function  $f : \{CM_1, CM_2, \dots, CM_k\} \times \Theta \rightarrow \mathbb{R}^j$  defined appropriately, for some parameter space  $\Theta$ . The function may use further parameters  $\theta$ . One example of such a function that goes beyond the concept or edge vector and is used in the case studies is a distance matrix of concept maps. The mapping function  $f : \{CM_1, CM_2, \dots, CM_n\} \times \{CM_1, CM_2, \dots, CM_k\} \rightarrow \mathbb{R}^k$  is defined as:  $f(CM_i, \{CM_1, CM_2, \dots, CM_k\}) = v_i \in \mathbb{R}^k$  such that the  $j$ -th element of the resulting vector  $v_{ij}$  is the graph similarity  $C(CM_i, CM_j)$  as defined in section 5.2.2. This matrix will be called the *graph similarity matrix* later on.

### 7.1.1.2 Horizontal

In contrast to vertical landscapes, a horizontal landscape, by definition, encompasses a temporal aspect. The simple notion of discrete time events as presented in section 6.2.1 is enough, however.

Modeling a horizontal amalgamation requires some form of temporal information for graphs. By the definition given above, an amalgamation is a graph where the maps forming the graph are no longer identifiable as single entities. A horizontal landscape, however, has a time component that must be kept. The concept landscape is then defined - in analogy to the concept maps based on dynamic sets - as a dynamic structure for a point in time  $i$ :  $CL(i) = (V(i), E(i), L_{v_i}, L_{e_i})$ , where  $L_{v_i} : V(i) \rightarrow L(i)$  and  $L_{e_i} : E(i) \rightarrow L(i)$  are defined as labeling functions that are now using dynamic sets instead of static ones.

Accumulating concept maps means that the constituent maps are still identifiable as single entities in the aggregation. Therefore, similar to the modeling of temporal aspects of a concept map, it can be defined as a dynamic set  $CL(i)$  of concept maps  $CM_1, CM_2, \dots, CM_k$  that form the accumulation, i.e.  $CL(i) = CM_i$  for all

$1 \leq i \leq k$ , where each  $CM_i$  is one of the  $k$  measurements that are used for the horizontal landscapes. In other words,  $CL$  defines a time series of concept maps. Additionally, it is convenient to define additional dynamic sets  $V(i), E(i), L_v(i), L_e(i)$  that map onto the corresponding part of the  $i$ -th concept map. For example if  $CL(i) = (V_i, E_i, L_{v_i}, L_{e_i})$ , then  $V(i) = V_i, E(i) = E_i$  and so on. Note that  $L_{v_i}$  and  $L_{e_i}$  by definition are sets of functions. Then, for example, the super set of all concepts used in the aggregation over all time is described by  $\bigcup_{i=1}^k V(i)$ , or the set of concepts that has been added between the two points in time  $i$  and  $j$  is given as  $V(j) \setminus V(i)$ , assuming that  $j$  is after  $i$  in the temporal ordering.

## 7.2 Analysis Methods

This section presents several ways of working with or analyzing concept landscapes. The methods are not new, but are adapted to work in this novel context. As has been noted above, the focus of the first method, cluster analysis, is on the inherent differences in the data by way of identifying sub-groups that differ in marked ways from one another. The focus of Pathfinder analysis presented next, in contrast, lies on the common structural elements of the data. Additionally, using graph measures and visualization techniques for concept landscapes are presented. Fig. 14 shows how the different methods relate to concept landscapes and how typical analyses might work.

The chosen methods have been used in the case studies and are also implemented in the software presented in the next chapter. All methods are inherently suited for computer based analysis. As the case studies will show, however, most can be gained by combining this automated analysis with a manual interpretation. The interpretation of the results is often based on quantitative (e.g. using a hypothesis to test the difference in concept map complexity between two groups of persons) as well as qualitative (e.g. visually inspecting and describing the differences in structure between two concept landscapes) analysis methods.

### 7.2.1 Cluster Analysis

A typical workflow of using concept landscapes in conjunction with cluster analysis is as follows:

1. Accumulate the concept maps (vertically) into a matrix using any desired mapping function.

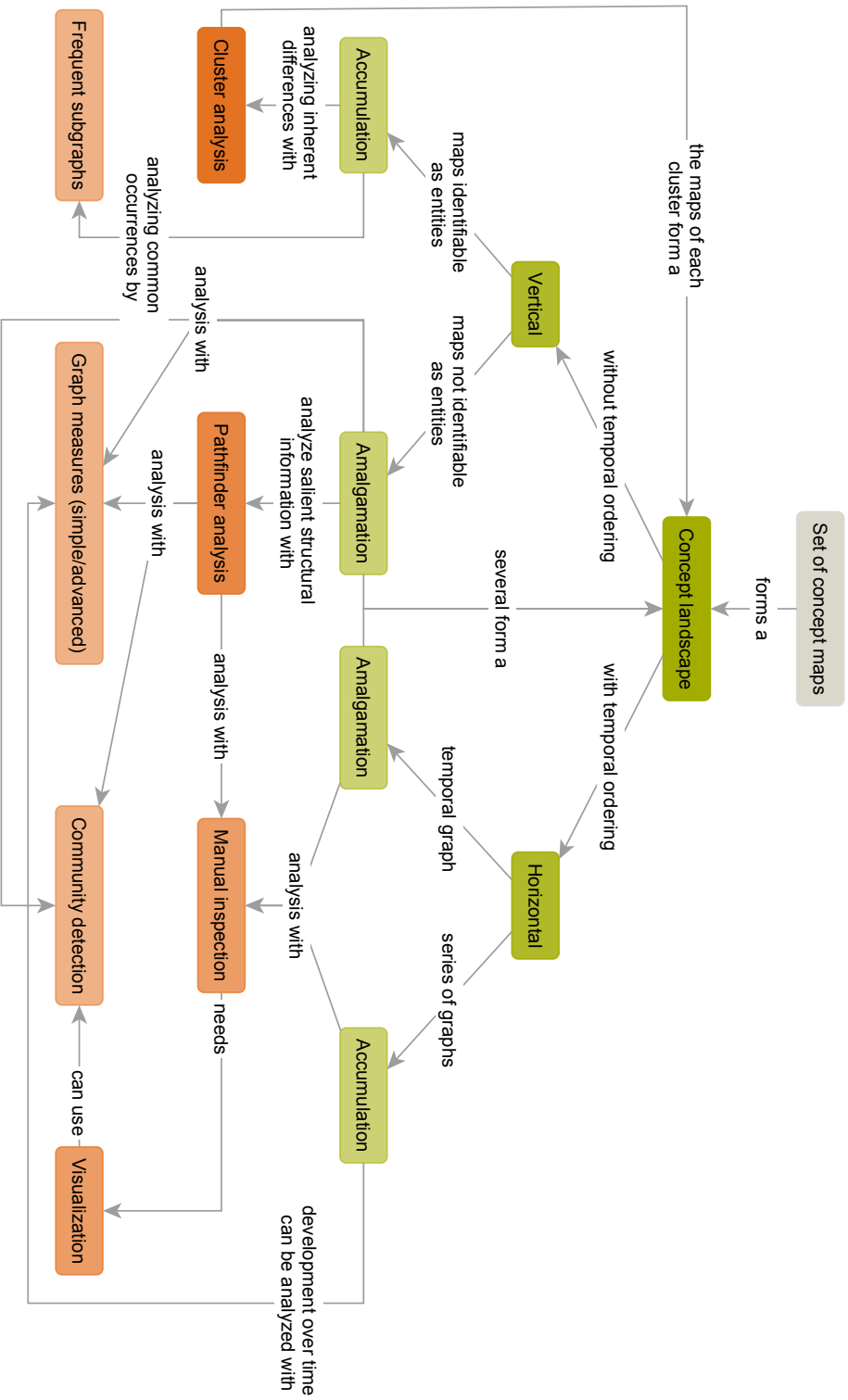


Fig. 14: Schematic overview over the different analysis methods and their relation to the different types of concept landscapes.

2. Apply a clustering algorithm to the resulting (often binary) matrix.
3. Analyze the quality of the clustering and possibly revert to the second step (using different parameters) until the solution can be considered locally optimal.
4. Analyze the clusters regarding their differences, for example by forming new landscapes of the maps of each cluster separately.

Concerning the first step, when combining the data of a set of concept maps, often the result will be a matrix of binary indicator variables - for example when using the concept matrix or edge matrix as defined above. As the results of the case studies have shown, using the edge matrix seems to give less satisfactory results than the concept matrix. This may be due to the sparseness of the edge matrix, which has  $\Theta(n^2)$  columns for  $n$  concepts. Clustering then requires either only very few concepts or very many concept maps, since the number of observations (maps) must usually exceed the number of dimensions (columns) to provide good results for clustering. Other methods of aggregation are possible as well, the best method will usually depend on the particulars of the data. Note that not all aggregations are useful for every clustering method. The graph similarity matrix, for instance, is mostly suited for distance based partitioning methods, as it effectively is the distance matrix of the observations. In a different context, Valerio et al. (2008, p. 125) give several features of concepts that could also be used to encode the map, like the number of concepts that can be reached from each concept or the number of incident edges for each concept.

The last step of analyzing the clusters is of major importance, as the clustering algorithms itself may or may not produce clusters that are actually worthwhile from a researcher's perspective. Also, it is important to note that in general the clusters can only be interpreted by additional analysis steps. Otherwise it may happen that the algorithm produces a clustering based on the influencing factors of the externalization that the concept landscape tries to minimize: If, for example, the maps of one cluster are very sparse, then maybe this cluster is formed by all the persons that didn't understand concept mapping or weren't motivated at the time of creation to produce larger maps. Since a clustering itself is nothing more than several sets of maps again (or in other words, concept landscapes), all techniques can also be applied to the clusters individually and the results compared. The case studies offer some insights into how this may be used to identify differences between clusters.

In the following, two methods of clustering are presented that are applicable for the second (and third) step and were successfully applied in experimental studies. One

uses a partitioning method and distance measure, the other uses a latent class approach. While not every clustering method is appropriate - like using Gaussian mixture models as shown below - there are clustering methods, like hierarchical clustering, that can be applied in theory but are not used here. The reason for this is that the partitioning methods and the chosen latent-class approach are by definition suitable for the task and are achieving the desired goal whereas, for example, a hierarchical clustering would yield information that can not directly be interpreted in the context of concept landscapes. It might still prove to be a useful clustering method, though. However, the case studies have shown that the chosen methods are giving the most promising results. But identifying an optimal algorithm for clustering concept maps, or comparing clustering algorithms in order to gain insights into their particular strengths and weaknesses on a larger scale is beyond the scope of this thesis. The algorithm that is used for clustering can (and should) be adapted or changed, if different methods are providing better results for a particular data set.

### 7.2.1.1 Similarity Based Clustering

Following the straight forward definition of clustering, a cluster should be comprised of elements that are “similar” to each other, yet “dissimilar” from the elements of the other clusters. Similarity-based clustering works exactly based on this notion. For concept landscapes, as noted above, the input data is often a binary matrix. So, before a similarity based clustering algorithm can be applied, a distance measure must be chosen. This measure will be applied to pairs of observations, i.e. pairs of binary vectors, and must yield a positive real value that is small for “similar” observations. These values are forming the distance matrix as defined in section 5.3.1. As noted in section 5.3.1, common choices are the Manhattan or Euclidean distance. Conceptually, for binary data resulting from concept maps, the Euclidean distance doesn’t have a straight-forward interpretation though. The Manhattan distance ( $L_1$  norm), usually defined as  $\sum_{i=1}^n |x_i|$  for an  $n$ -dimensional vector  $x$ , applied to the difference of two binary vectors effectively only encodes the number of positions in which the two vectors differ. This makes sense for, e.g. the concept vectors of two concept maps: Clearly two maps that share concepts are more similar than two that don’t. On the other hand, at least when using the concept matrix, much structural information of the maps is lost and thus can’t be incorporated in calculating the similarity. Using the graph similarity matrix doesn’t suffer from this problem and shows interesting results as shown in the second and third case studies in chapters 11 and 12 respectively.

Concerning the clustering algorithm itself, every one is appropriate in principle, as long as it doesn't pose restrictions (or assumptions) on the input data that cannot be met. As noted in section 5.3.1, k-means clustering implicitly assumes that the Euclidean norm is valid for the given data, for example, which is usually not the case for concept landscapes. Apart from such restrictions, the algorithm itself is less important than the actual distance measure though, since only this measure imposes how "different" or "similar" single observations are. Different similarity based clustering algorithms should therefore arrive at similar results when using the same measure of distance. In the experiments in this section and in the case studies only the k-medoids algorithm as presented in section 5.3.1 will be used in the form of its most prominent implementation PAM. As has also been noted in section 5.3.1, to ensure that a chosen number of clusters is optimal, both the non-uniformity of the data and the quality of a found clustering must be taken into account; the Hopkins index and the G1 index as defined there are used throughout this thesis for these tasks.

### 7.2.1.2 Latent Class Clustering

This section presents an alternative approach to using distance measures and similarity for clustering and is based on a probabilistic model of the data. Such a model must be able to generate the observations used for clustering. As has been noted in section 5.3.2, one advantage of this approach is the possibility to quantify the probability of different model parameters. This can be used for model selection which, in particular, encompasses the sub task of identifying the best number of clusters.

What probabilistic model can be used in order to describe an accumulated set of concept maps? At least for the case of using the concept or edge matrix, the data is a binary matrix with rows corresponding to concept maps and columns encoding the presence or absence of a particular feature of the map. So, there are a number of observations of a number of binary variables and often there also will be correlations among the variables. For example for a concept vector, a 1 in one column will always indicate the presence of another 1, as each edge has two incident concepts. One possible model is built upon the Bernoulli distribution. The basic definitions and the derivation of the clustering in the next paragraphs can be found for example in (Wolfe 1970) and (Stibor 2008).

A Bernoulli distribution with parameter  $p$  models the probability of a random experiment with two possible outcomes. One outcome has probability  $p$ , the other has

$1 - p$ . The probability function is  $P(x) = p^x(1 - p)^{1-x}$ , where  $x \in \{0, 1\}$ . A vector of  $k$  binary random variables can be modeled with a multivariate Bernoulli distribution of length  $k$ . In this case, the parameter  $\Theta$  consists of  $k$  values  $(p_1, p_2, \dots, p_k)$ . The probability of an observation  $x = (x_1, x_2, \dots, x_k) \in \{0, 1\}^k$  is given by the function  $P(x) = \prod_{i=1}^k p_i^{x_i} (1 - p_i)^{1-x_i}$ . To denote the dependence of  $P$  on its parameter, it's customary to use the notation  $P(x|\Theta)$ .

A multivariate Bernoulli distribution does not allow correlations among the dimensions of the observations. Each dimension is independently assumed to take on a value according to this dimensions probability. Following e.g. Stibor (2008), higher order correlations can be included by extending the model to a mixture of multivariate Bernoulli distributions. So, instead of having only one multivariate Bernoulli distribution, a set of such distributions is used to model observations. For clustering, each of the mixture's components is taken to represent one cluster. Every observation in turn has a probability of "belonging" to each cluster. A multivariate Bernoulli mixture model (MBMM) can be used to model concept map data. By finding the most probable values of its parameters, a clustering can be determined. See the example below for a demonstration of the capabilities regarding the identification of correlations.

The parameters of a MBMM consist of  $m$  vectors  $\Theta = (\Theta_1, \Theta_2, \dots, \Theta_m)$  with each vector describing a single multivariate distribution and  $m$  mixing coefficients  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$ . It holds that  $\sum_{i=1}^m \alpha_i = 1$ . The  $\alpha_i$  encode the contribution of each distribution to the clustering. Each  $\Theta_i$  consists of  $k$  values, so a MBMM has  $m \cdot k + m - 1$  parameters as there are  $m - 1$  mixing coefficients that are free to choose, the last one is completely determined by the others. These must be chosen (optimally) to identify a clustering. The probability function for an observation  $x = (x_1, x_2, \dots, x_k) \in \{0, 1\}^k$  is  $P(x) = P(x|\Theta, \alpha) = \sum_{i=1}^m \alpha_i P(x|\Theta_i)$ .

Finding the optimal parameter values for given input data corresponds to maximizing the likelihood of the observed data. As is customary, instead of maximizing the likelihood, the log-likelihood is maximized, as it is numerically more convenient to sum large values than to multiply small ones. The log-likelihood of  $n$  observations  $X = x_1, x_2, \dots, x_n$  is

$$L(X|\Theta, \alpha) = \log\left(\prod_{i=1}^n P(x_i|\Theta, \alpha)\right) = \sum_{i=1}^n \log(P(x_i|\Theta, \alpha))$$

As there is no analytic solution to this maximization problem, the EM algorithm presented in section 5.3.2 is used to find locally optimal values. For a model built on multivariate Bernoulli mixtures, the E- and M- steps are:



**E-Step** Calculate for each mixture component  $m$  and each observation of the data  $x_i$  the posterior probability using Bayes theorem:

$$P(m|x_i, \Theta, \alpha) = \frac{P(x_i|m, \Theta, \alpha)P(m)}{P(x_i)}$$

**M-Step** Calculate for each mixture component  $m$  new parameter values  $\Theta'_m, \alpha'_m$  using the current values  $\Theta_m$  and  $\alpha_m$ , by:

$$\alpha'_m = \frac{1}{n} \sum_{i=1}^n P(m|x_i, \Theta, \alpha)$$

and

$$\Theta'_m = \frac{1}{n\alpha'_m} \sum_{i=1}^n P(m|x_i, \Theta, \alpha)x_i$$

The computation can be effectively implemented using matrix operations. The output of the EM-algorithm consists of the locally optimal (i.e. most probable) values of the probabilities  $\Theta$  and mixing coefficients  $\alpha$ . These values can be used to calculate the posterior probability (as given above) for each observation and each cluster. This, in other words, is the probability for each observation to belong to any of the clusters. Usually, the maximal value of these probabilities is then taken as the cluster this observation is assigned to. However, the probability values can also be used to model uncertainty in the cluster assignment or to filter out observations below a given threshold of probability for any of the clusters.

The clustering process performs the EM-algorithm for a range of different number of components and chooses the solution with the lowest AIC value (see section 5.3.2). This will then select not only the most probable parameters of the MBMM model, given the observations, but it will also select the most probable number of components (clusters). The MBMM clustering approach is employed in the second and third case study presented in the next part.

### 7.2.1.3 Example

To investigate how the two clustering approaches work, two examples using randomly generated, artificial data are presented here.

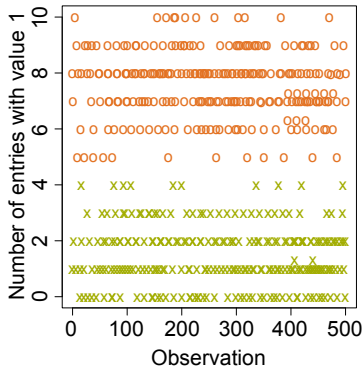
For the first example, the data consists of a matrix of binary values with 500 rows and 10 columns, representing, e.g. the concept matrix of a vertical accumulation of 500 concept maps that were restricted to a common set of 10 concepts. Typically, some of these concepts will appear in many maps, as they are, for example, “basic”

or often used concepts of the subject matter. Conversely, there will be concepts that appear only in few maps, for example because those concepts are more “advanced” or seldom used. Similarly, one expects to find some persons that only use few concepts in their maps and persons that use more or nearly all concepts. Finding clusters that show these differences between the concepts or between the persons is one of the goals that cluster analysis should achieve.

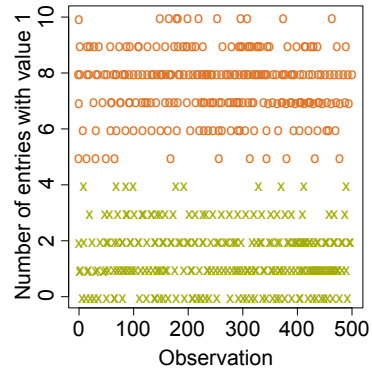
The random data is generated to reflect both the structural differences between concepts, as well as the differences between persons. Each (artificial) observation is drawn from one of two multivariate Bernoulli distributions with 10 dimensions. The order is random, but there are 250 observations drawn from each distribution. Both distributions have increasing probabilities for the dimensions, the first ranges from  $p_1 = 0.025$  to  $p_{10} = 0.25$  and the second from  $p_1 = 0.525$  to  $p_{10} = 0.75$ . So, the data is generated such that it naturally consists of two clusters, one with a rather low probability for each dimension and one with a higher probability, but also reflecting the same increasing probabilities between the concepts.

A clustering algorithm, when applied to this matrix, should identify the two clusters and it should also be possible to show that two is the optimal number of clusters. Both the PAM algorithm using Manhattan distance and the MBMM clustering were applied. The experiment was repeated several hundred times with new randomly generated input to ensure that the results were not by chance. The Hopkins index applied to the data always yielded values at or above 0.9, clearly indicating the non-uniformity of the data set. The results of the clustering algorithms stayed predictably constant, in the sense that the optimal number of clusters was always two and the clustering itself followed more or less exactly the generating model. Fig. 15 shows the results of both algorithms for one of the runs. For this artificial data set, both PAM and MBMM clustering were able to accurately capture the underlying structure.

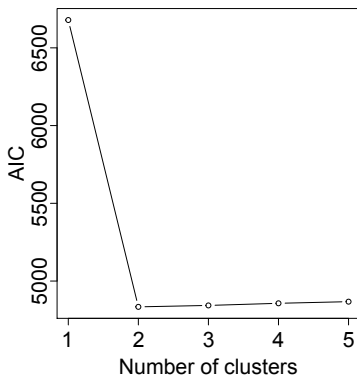
Fig. 15(a) and Fig. 15(b) show the result of the clustering itself by using a different character and color for each observation, depending on its assigned cluster. An observation is plotted as a point, using the number of dimensions that have a 1 as the plotted value. For the MBMM approach, the mixture component with the highest probability of generating the observation was used as the cluster. The two clusters are visually identifiable as the dense regions at the bottom (low probability for each dimension) and the top (high probability for each dimension). Both PAM and the MBMM clustering identify the clusters “correctly” in the sense that the clusters are comprised exactly of the 250 observations generated for each distribution. The extreme cases of vectors with all zeros or all ones are identified as the medoids by the PAM algorithm. Fig. 15(c) and Fig. 15(d) show that two is the optimal number of clusters, when disregarding more than five clusters.



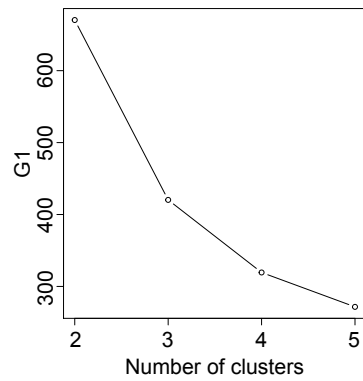
(a) MBMM clustering



(b) PAM clustering



(c) AIC values



(d) G1 values

Fig. 15: MBMM and k-medoid clustering for the dataset of the first experiment in comparison

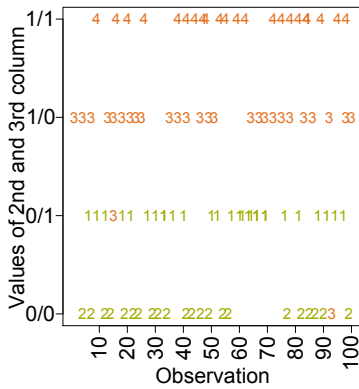
The EM-algorithm was set to stop as soon as the absolute change in value for any of the parameters in one iteration was less than 0.0001. For the plotted run eight iterations were made. The likelihood of each point to belong to the chosen clusters is less than 0.9 for only eight observations. Five of those are from the fringe. The mixing coefficients in the end were both 0.5. The probabilities of the distributions are estimated to range from 0.02 to 0.24 for the first cluster and from 0.58 to 1.00 for the second. So, the values don't represent the original probabilities perfectly, but the basic trend is clearly captured by the algorithm.

For the second example, the clustering should find more indirect dependencies in the data. As has been noted above, for concept vectors the single dimensions are not strictly independent, as each edge has two incident concepts. In a real world task, it seems reasonable to assume, that this dependency will not spread uniformly over all dimensions for each map. Instead, several combinations will be more probable than others, as typically a concept will often only be connected with very few other concepts in a set of maps. An exploratory analysis that can retrieve such "patterns" provides interesting insights, therefore.

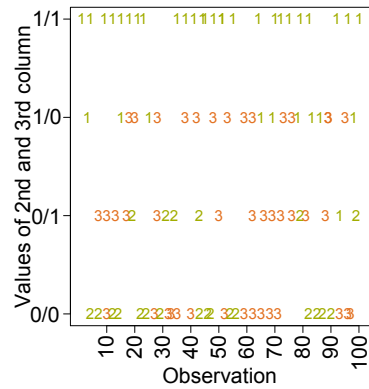
So, the data for the second example consists of 100 observations (rows) of five dimensions (columns) of binary values. Again, this can be taken to represent the concept matrix of a vertically accumulated concept landscape. This time, instead of randomly generating each column independently, the columns are made statistically dependent on each other. Specifically, the first three columns are created as uniformly distributed binary values. The fourth column is identical to the second column, indicating a statistical dependence (perfect correlation). The fifth column is the negation of the third column, i.e. derived by subtracting the third column from 1 for each row, again indicating a statistical dependence (perfect negative correlation). To make the data more realistic, noise was added to make the correlations less perfect by randomly choosing two observations and negating the fourth and fifth column, corresponding to a "noise level" of 2%.

Optimally, the clustering algorithms should identify these dependencies in the data. To correctly capture them, four clusters are needed based on the combination of values of the 2nd and 4th and the 3rd and 5th column. Of the 16 possible combinations that the four binary variables could have, only four are actually appearing, as the value of the 2nd and 3rd column determine the other two values. Again, PAM using Manhattan distance as well as the MBMM approach were used and the experiments were repeated in order to ensure that the results were not by chance.

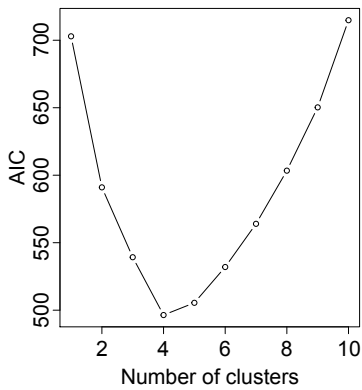
In this example, only the MBMM approach is able to capture the underlying information. Fig. 16 shows the results. While the optimal number of four clusters is clearly



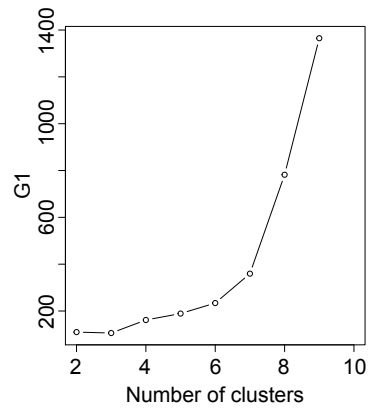
(a) MBMM clustering



(b) PAM clustering



(c) AIC values



(d) G1 values

Fig. 16: MBMM and k-medoid clustering for the dataset of the second experiment in comparison

determined using the AIC values and MBMM, it is not as clear-cut when using PAM and the value of G1. As shown in Fig. 16(a) and Fig. 16(b), the MBMM clustering is able to capture these dependencies perfectly, whereas PAM clustering is oblivious to the structural dependencies. When looking at the learned parameters for the four mixture components, the following values have been identified after only 10 iterations of the EM algorithm:

	Column 1	Column 2	Column 3	Column 4	Column 5
Component 1	0.46	0.00	1.00	0.00	0.00
Component 2	0.27	0.00	0.00	0.00	1.00
Component 3	0.68	0.93	0.04	1.00	0.96
Component 4	0.36	1.00	1.00	1.00	0.00

It clearly can be seen how the noise is affecting the parameters in the third component and how the dependencies have been captured perfectly: Aside from noise, the second and fourth column are showing the same values; the fifth column is showing the negated value of the third column.

To underline the importance of a valid model for the input data, when using a clustering that is based on Gaussian mixture models, i.e. assuming that the observations are generated from a number of Gaussian distributions instead of Bernoulli distributions, the clustering with the most probable fit according to BIC, consists of 169 clusters for the first and 95 clusters for the second example. The test was done using the package *MClust*<sup>1</sup> for R.

These two examples show that using MBMM clustering is superior for concept matrices, since it is able to capture different aspects of hidden structural information. However, it is the task of the researcher to actually identify what the hidden structural information in the clusters is. Therefore, it may be well worth the effort to use different clustering algorithms on the same data and check for differences and similarities in the identified clusters. If, for example, PAM and MBMM show clearly varying results, it may be worthwhile to check for hidden dependencies between the variables of the observations. If both are more or less finding the same clustering, then this may indicate that the structural information in the data is simpler. One advantage of PAM is, that it can also be used with other different measures, like the graph similarity matrix.

<sup>1</sup><http://www.stat.washington.edu/mclust>

## 7.2.2 Pathfinder

From its original intended use, as described in section 5.2, a Pathfinder network helps with identifying the salient information prevalent in multi-dimensional similarity ratings. The lengths of paths in the network contain information about how “close” or similar the connected concepts are in the data. From a graph theoretical point of view, constructing a Pathfinder network is simply an algorithmic method of (edge-)pruning a graph by keeping all nodes and systematically removing edges. Taken together, this is also the reason why the Pathfinder algorithm was chosen as a suitable analysis method for concept landscapes: On the one hand, in contrast to other scaling techniques like MDS, it can work directly on a graph as input and it also produces a graph, making it suitable for the particular format of concept landscapes. On the other hand, its original intention is directly related to analyzing the structure of conceptual knowledge, in contrast to other graph pruning techniques, like minimal spanning trees or the simple edge removal based on some threshold as for example in (Larraza-Mendiluze & Garay-Vitoria 2013). This makes it suitable for the particular information contained within concept landscapes. Even though strictly speaking, a concept landscape is not a “network” in the psychological sense that is underlying the term “Pathfinder network”, the terms “Pathfinder network”, “Pathfinder analysis” or simply “Pathfinder” are used synonymously throughout this work.

In contrast to cluster analysis, Pathfinder analysis works by explicitly discarding information of the original data that is considered not salient enough. When applying the Pathfinder algorithm to a concept landscape, the goal is to gain insight into the “common” structural elements of the concept maps forming the landscape. In other words, if two concepts are connected in the resulting graph, it should be possible to assume that these two concepts are “typically” seen as connected, given the original concept maps. Therefore it is best not to think about “similarity” in the context of this thesis, but to instead think of “connectedness”. Given a graph (concept landscape) that contains measures of the frequency of connectedness of pairs of concepts, the Pathfinder algorithm identifies the salient structural information regarding the most frequent connections between any pair of concepts. This information of the common connections is of course already present in the landscape. Further analysis is only needed in order to make it visible, as typically the combination of a set of concept maps will contain numerous structural elements, some appearing in almost every constituent map and some only appearing in very few, or even just a single map. An analysis using concept landscapes with Pathfinder networks is typically following these steps:

1. Amalgamate the maps vertically, using an appropriate method for defining the edge weights.
2. Create the Pathfinder network from the graph generated in the first step.
3. Analyze the generated Pathfinder network manually or automatically, e.g. by using graph measures.

An appropriate definition of the edge weights must somehow reflect its “commonness”. Using the simple summing of edges as defined in section 7.1.1 above cannot be used, as the Pathfinder algorithm expects a smaller edge weight to indicate a more favorable edge. Therefore, a transformation of weights is necessary. It is not directly obvious, though, that the Pathfinder algorithm will work as expected, since such a transformation is closely related to the problem of finding simple paths of longest weights in graphs, which has been shown to be  $\text{NP}$ -hard (cf. Lawler 2001, p. 9). So, unless  $\mathbb{P} = \text{NP}$  there is no polynomial transformation of a graph’s edge weights such that searching for shortest paths with the transformed values would equal a longest path with the original values. The Pathfinder algorithm however, while closely related to the problem of finding shortest paths, does work in a different way and a transformation is possible as the next paragraphs show.

A way that keeps the linear distances of the edge weights the same and also keeps the range of numbers the same, is to simply use the number of concept maps plus one and subtract from this the number of concept maps in which the edge is present. Formally, the weight function of the amalgamation  $w : E \rightarrow \mathbb{N}$  is defined as  $w(e_i) = k + 1 - |\{1 \leq j \leq n | e_i \in E_j\}|$  for a combination of  $k$  concept maps. An edge that appears in every map is assigned a weight of 1 and an edge that appears in only 1 map is assigned a weight of  $k$ .

As noted in section 5.2.1, the Pathfinder algorithm will make the graph  $q$ -triangular: An edge is removed if and only if there is a path of length at most  $q$  with a smaller distance (using the  $r$ -metric) than the weight of this edge. To see that the transformation does what it is intended to do at least for the two cases  $r = 1$  and  $r = \infty$  that will be used exclusively, assume that an edge  $e_i$  is removed by the algorithm. This means that must exist a path of length at most  $q$ ,  $e_1, e_2, \dots, e_q$  such that (for  $r = 1$ ) the inequality  $w(e_1) + w(e_2) + \dots + w(e_q) < w(e_i)$  holds. Without loss of generality, the length of the path can assumed to be exactly  $q$ . As  $e_i$  is removed, it must violate the  $q$ -triangularity. Let  $c_i$  be the number of maps in which edge  $e_i$  is present. Then  $w(e_i) = k + 1 - c_i$  for some constant  $k$ . It follows that:



$$\begin{aligned}
k + 1 - c_1 + k + 1 - c_2 + \dots + k + 1 - c_q &< k + 1 - c_i \\
qk + q - (c_1 + c_2 + \dots + c_q) &< k + 1 - c_i \\
(q - 1)k + q - 1 - (c_1 + c_2 + \dots + c_q) &< -c_i \\
-(c_1 + c_2 + \dots + c_q) &< -c_i \\
c_1 + c_2 + \dots + c_q &> c_i
\end{aligned}$$

In other words, there exists a path with a combined sum of “edge occurrences” that is greater than the number of “edge occurrences” of the edge that is removed. By the same reasoning,  $r = \infty$  works as well, by simply taking the maximum weight instead of the sum in the inequality.

There are other ways of defining the edges’ weights of course. For example, if scores are present, it seems reasonable that the identification of “common” elements of knowledge (or misconceptions for that matter) should be based only on edges that have a certain score.

As has been noted before, a Pathfinder network will always contain the same components and the same nodes as the original graph. Therefore, if for example, a concept is used in only one map of the aggregation, this concept will remain in the Pathfinder network using the one edge that is present in the data, because pruning this edge would create a new component. Clearly, however, this can hardly be taken as an indicator that this edge is representative of the “common” knowledge structure, as it is only there as an artifact of an idiosyncratic knowledge construct that is not removed by the analysis method.

So, it is important to filter out these idiosyncrasies by manual removal. There are different ways, with different results. In general, either nodes or edges can be removed and this can happen either before or after the Pathfinder algorithm. Removing concepts amounts to summing the rows or columns of the weight matrix of the amalgamation and removing those, whose value is higher (using the transformed weights) than some chosen threshold. This first establishes a set of “common knowledge” concepts and then searches the most frequent structural connections between them. Removing edges amounts to setting all entries in the weight matrix to  $\infty$  that are higher than some chosen threshold - this may leave concepts unconnected. In contrast, this first establishes a common set of propositions and then finds the most common arrangement of concepts. The second approach is stricter, in the sense that it will, given the same threshold, always leave a superset of the concepts unconnected that would be filtered out by the first approach. In the case studies, both approaches were used. In these studies, however, there was not

much difference, if at all, between the methods. There is no best way to arrive at a threshold for filtering, but using percentiles or a gap statistic provide reasonable approaches. If there is a value where decreasing the threshold a bit further doesn't lead to more removals, this can be seen as the "gap". Generally, it seems advisable to try several threshold values and inspect the effects on the network.

The creation of a Pathfinder network itself is dependent on the choice of the parameters  $q$  and  $r$ . In general, higher values for either parameter will produce sparser networks. As has been mentioned in section 5.2.2, it is advisable to try several values and judge the quality of the results. Also an argument against higher  $q$  values has been presented there. This is only valid in the context of assessing similarity data. From a perspective of graph pruning, the typical choice for the value of  $q$  is  $|V| - 1$ . Using smaller values would be useful only if there were a theory about why the structural information of paths that are exceeding a certain number of intermediate steps is not as important as the information of shorter paths. For  $r$ , the extreme values of 1 and  $\infty$  are most appropriate. Euclidean distance ( $r = 2$ ) might be interesting if it can be assumed that the spatial placement of concepts in a map is indeed an important aspect of the organization of knowledge. If an appropriate aggregation is used that somehow respects this spatial information, the Pathfinder network with  $r = 2$  may produce relevant insights. Other values in between these extreme cases might be seen as becoming gradually more and more like the boundaries but generally there is no theory about the effect of using these values.

The pruning of the Pathfinder algorithm differs fundamentally from a manual removal of edges with a low (or high) weight. Therefore, it is paramount for the analysis of a Pathfinder network to keep in mind that it is in general not possible to compare, for example, two different pairs of concepts based on the fact that the edge between one of these pairs was removed by the Pathfinder algorithm, but not between the other. If a concept, for example, has only one incident edge, this edge will never be pruned. If the weight of this edge is high, it may happen that every edge that is pruned by the Pathfinder algorithm actually has a lower weight than this one. So, the manual filtering must make sure that every edge is "common" enough (in the given context) before applying the Pathfinder algorithm. Also, it may be advisable to keep the actual edge weights in mind for analysis and only focus on the "strongest" edges for interpretation.

### 7.2.2.1 Example

As before with the clustering algorithms, an example using artificially created data is used to illustrate how the Pathfinder analysis of concept landscapes works. Real world applications can then be found in the case studies presented in the next part.

The example is based on an assumed concept mapping survey using the concepts: *class*, *object*, *attribute*, *method*, *visibility* and *algorithm*. There are 10 concept maps based on this list shown in part A of the appendix. Amalgamating them vertically results in the graph shown in Fig. 17. The edge weights are formed by using 11 minus the number of concept maps that contain the given edge. A filtering in this case is unnecessary, as it is densely connected. Removing for example all edges with a weight of 10 (lowest quartile of the original edge weights), or even all edges with weights down to 7 (lower half of the original edge weights) would neither leave a concept unconnected nor increase the number of components.

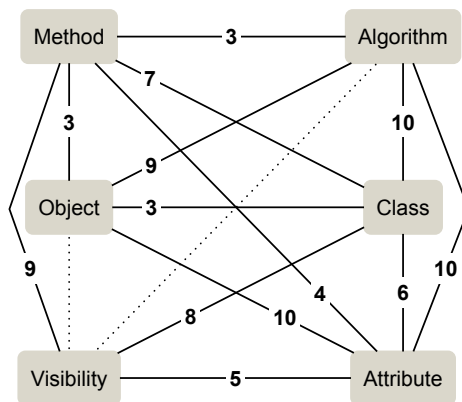


Fig. 17: The amalgamated concept landscape of 10 concept maps. The edge weights are transformed by using 11 minus the number of maps in which the given edge is present. The dashed edges are non-existent in the original concept maps and thus have an assumed weight of  $\infty$ .

Using the Pathfinder algorithm on this graph with the two most prominent values for  $q$ , namely  $q = 2$  and  $q = n - 1 = 5$  and the values 1 and  $\infty$  for  $r$ , gives four different resulting networks shown in Fig. 18(a) to Fig. 18(d) respectively. As is expected, the number of remaining edges decreases both when increasing  $q$  or when increasing  $r$ , or both. When taking a closer look at how the edges are pruned, several things are worth noticing:

- Each Pathfinder network contains the three “strongest” links (i.e. smallest weights) between *class*, *object*, *method* and *algorithm*. These links clearly can be seen as important when manually inspecting the input graph, too.
- Not every Pathfinder network has pruned all of the three weakest links (*algorithm* - *class*, *algorithm* - *attribute* and *attribute* - *object*), but the sparser networks are not containing them anymore and two of the three have been removed in all networks. In general, higher values of  $q$  will foster the removal of weak links, since there are more possible alternative paths that are inspected by the algorithm.
- The sparsest graph is produced by  $q = 5$ ,  $r = \infty$  and is also a minimal spanning tree of the graph.
- The difference between setting  $r = 1$  and  $r = \infty$  is obvious for example for the link between *method* and *visibility*: As each alternative path, when summed, weighs more than the direct edge, it is never pruned for  $r = 1$ . However, with a weight of 9 it doesn't seem to represent a “common” understanding shown in the set of maps.

Concerning the interpretation of the graphs, it is important to note that even though the Pathfinder algorithm is actually inspecting paths, the interpretation of the pruned networks is centered around links. Taking, for example, the connection between *algorithm* and *attribute* that has been pruned in each of the Pathfinder networks of this example: It is questionable to say that this means that the concept *algorithm* is seen as related to the concept *attribute* only through the concept of *method* - as the concept maps themselves are not containing information that extends beyond a single edge. However, it is valid to say that, in general, the concepts *algorithm* and *attribute* are not seen as related, while the concepts *algorithm* and *method* as well as *method* and *attribute* are.

Following the observations above, it seems advisable to set  $q$  to the highest value when using concept map data and also to use  $r = \infty$ . This will produce the sparsest graphs showing only the strongest connections that are needed to make sure the graph is still connected. Using the maximal value ensures that every alternative is inspected for pruning and no artificial cut-off length is used. With the value of  $q$  fixed, the value of  $r$  can then be seen simply as gradually pruning more and more edges. Depending on the analysis that is intended afterwards, either keeping more edges and thus setting  $r = 1$ , or pruning as much edges as possible by setting  $r = \infty$  is possible.

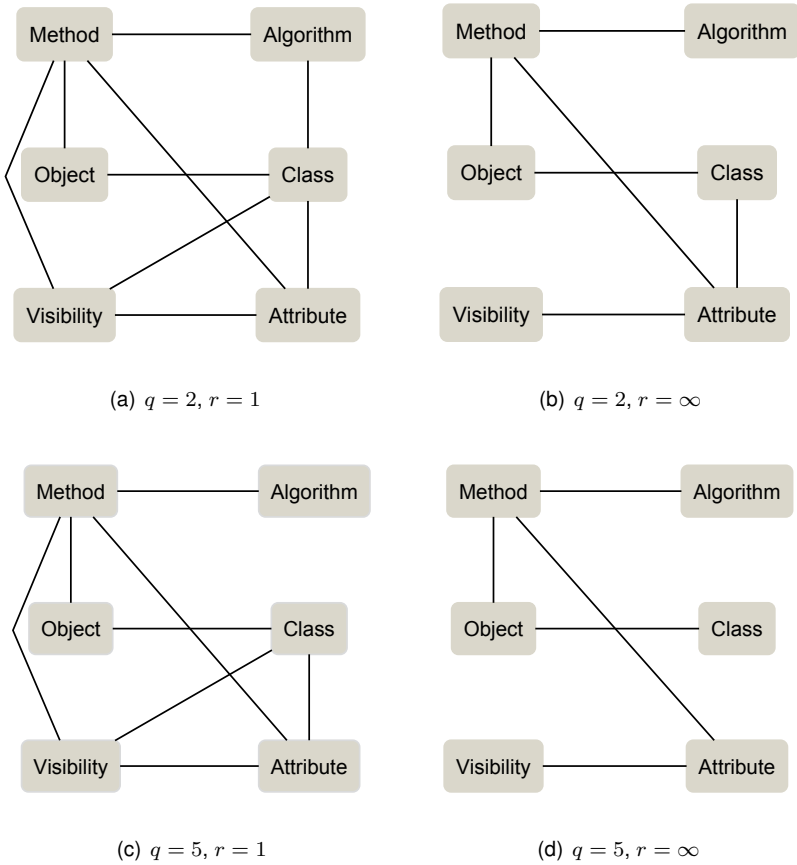


Fig. 18: Pathfinder networks created from the input graph in Fig. 17 using four different sets of parameters.

### 7.2.3 Graph Measures

For amalgamations the resulting data structure is, by definition, a graph. Therefore, it is possible, of course, to use graph measures for analysis, that are regularly used in the analysis of single concept maps (cf. Mandl & Fischer 2000, p. 5); for example by counting the propositions. However, as the measures are simply a different (usually numeric) representation of attributes of a graph, they should only be employed when there is some theoretical reasoning about why a certain attribute is informative for the task at hand. Measuring attributes of graphs is especially useful when comparing aggregations, for example in order to identify differences between the concept maps of clusters that were identified in the data. Also, they can be used in order to gain further insight into Pathfinder networks.

The following sections present - in no particular order and without claiming completeness - some useful measures. Leake et al. (2005) for example, investigate several different graph measures specifically for identifying important concepts of a concept map. These are investigated in more detail by Valerio et al. (2008). Most of the measures presented here are applied in the case studies in the next part.

#### 7.2.3.1 Simple Graph Measures

Clearly, the simple number of concepts and edges can be used as indicators for the analysis. As shown in section 4.2.2, scoring schemes based on the number of propositions are commonly employed for concept maps. More (correct) edges are, by the basic nature of concept maps, an indicator for a more densely connected knowledge structure. More concepts can be seen as an indicator of the breadth of the knowledge structure. Instead of using the number of edges directly, the ratio of edges to concepts may also be used. In other words, instead of measuring the number of concepts and the number of edges, it may be more appropriate to measure the number of concepts (breadth of knowledge) and the density of the propositions (connectedness of the knowledge). Also, the number of components of the graph can be measured. This is, for example, done by Ifenthaler (2006) under the term “ruggedness”. While typically a less “rugged” concept map is seen as better, the knowledge-as-elements view of conceptual change offers positive support for a more rugged map as not necessarily worse than a completely connected map.

For horizontal landscapes, often the development of the measures taken over time are interesting for the analysis. If concept map data has been combined first vertically and then horizontally, for example mean and standard deviation or

percentiles of the measures can be plotted/inspected as shown in the first case study.

For vertical landscapes, one is often interested in identifying the differences between groups, for example when using cluster analysis first and then aggregating the maps of each cluster separately. The simple difference between groups is then usually not enough for interpretation. Instead, statistical tests should be used in order to find out whether or not the difference is significant and not just a random artifact of the particular data. For example, for the number of concepts and two groups, a *t-test* can be employed in order to find out whether or not the two groups have a significantly different mean of the number of concepts. This requires that the two groups are independent samples, which they typically are and that the number of concepts is normally distributed with equal variance between the groups. This can be tested with additional tests. Usually normal distribution can be assumed if the groups are large enough.

### 7.2.3.2 Advanced Graph Measures

There is a range of different graph measures that can be used. A selection will be presented here with a short description and an explanation of why this particular measure can be assumed to work well for concept maps/landscapes. The second case study uses several of these measures.

**Centrality** The *centrality* of a node is a measure originating from research on social (and communication) networks and has been applied in the analysis of structural knowledge by e.g. Glöggler (1997). Also Larraza-Mendiluze & Garay-Vitoria (2013, p. 68) point to the “similarities between concept maps and social networks”. As Freeman (1978) points out, there are several concurring definitions of what a “central” node might be. The centrality measure of a node can be based upon its degree, the number of shortest paths that this node lies upon or the sum of minimal distances from this node to all others (cf. Freeman 1978, p. 219). All measures of centrality share the common behavior that the central node of a star is assigned the maximum centrality value of this graph and all nodes in a fully connected graph are assigned the same centrality value. Also, for each of the three methods, Freeman (1978) gives a method of calculation that is independent of the size of the graph, in order to allow comparisons between graphs of differing size. For concept maps, using the degree of a concept certainly yields a valuable insight as a higher degree can be seen as better connected in memory. The

*betweenness-centrality* of Freeman (1978) of a node or an edge is defined based upon the number of shortest paths in the graph that are using this node or edge. In general, a “central” concept of a concept map or concept landscape should be one that is fundamental for the knowledge structure, i.e. the connections of other concepts depend on the central concept.

**Connectivity** The *edge-connectivity* is the minimal number of edges between two given nodes that must be removed in order to separate the two nodes in the graph (cf. Balakrishnan & Ranganathan 2012, p. 53). Clearly, a high edge-connectivity between two concepts can be seen as an indication of a strongly developed structural knowledge surrounding these concepts. Conversely, a low edge-connectivity means that the concepts are only connected by a few paths and can easily be separated. Accordingly, the *graph-connectivity* is the minimum of all edge-connectivity values between all pairs of nodes. A high graph connectivity means that each pair of nodes has a high connectivity (which would typically be assumed for the densely connected knowledge of an expert. A low graph-connectivity only means that at least one pair of nodes has a low connectivity and is, as such, more difficult to interpret in a useful way.

**Diameter** The *diameter* of a graph is the longest simple (or shortest, if the graph has weights) path between any pair of nodes in the graph (cf. Ifenthaler 2006). In other words, without artificially prolonging the path by using circles or “expensive” edges, the diameter is the longest path that can be found in the graph between any two nodes. A linear, chain-like graph with  $n$  nodes and no weights will have a large diameter of  $n - 1$ , whereas a completely connected graph without weights will always have a diameter of 1. So, a small diameter is preferable, in general, as the concepts are then more densely connected, in contrast to Ifenthaler (2006, p. 48) who wrongly attributes a large diameter to a more complex graph. The diameter should also best be seen in relation to its maximal possible value as the absolute values of different graphs are usually not comparable - unless, for example, the number of nodes is held constant.

**Node Degrees** The *degree* of a node is the number of neighbors of the node. Used by itself, it can be taken as an indicator of how well connected a concept is in a knowledge structure. Additionally, when taking the degrees of all the nodes into account - also called *degree sequence* (cf. Balakrishnan & Ranganathan 2012, p. 10) - it can serve as an indicator of the three types of morphology “spoke”, “chain”,



and “net” as described in section 4.1.1. Specifically, if the average node degree is low - for a connected graph with more than two nodes, the minimal value is between 1 and 2 - and there is not much variation, the graph will resemble a “chain”. If the average node degree is high - the maximal value if the number of nodes less one, for a complete graph - and there is not much variation, the graph will resemble a “net”. Finally, if the average node degree is neither particularly low or high and there is variation (few nodes have a high degree, many have a low degree) the graph will resemble a “spoke”. For the prototypical concept maps of the three types, as shown in Fig. 6, with  $n$  nodes, the average node degree for a chain map as well as a spoke map is  $2 - \frac{2}{n}$  and for a (completely connected) net map it is  $n - 1$ . The standard deviation for the chain is approaching zero, as the number of nodes is growing. The standard deviation for the net is zero if each node has the same degree and the standard deviation of a spoke is growing with the number of nodes.

### 7.2.3.3 Community Detection

*Community detection* is a form of partitioning (or clustering) the nodes of a graph. A *community* is a subset of nodes such that the nodes within a community are densely connected, whereas the nodes of differing communities are not or less densely connected. The components of a graph are a natural way of forming communities. However, there are also methods that identify communities within a connected graph yielding new insights into the structure. Interpretation-wise, a community in a concept map can be seen as a set of concepts that are “belonging” together. There are several algorithms that are trying to identify communities based on different assumptions about how a community can best be characterized. Among others, the algorithms called “Walktrap” (Pons & Latapy 2006), “Spinglass” (Reichardt & Bornholdt 2006), and a greedy algorithm described by Clauset, Newman & Moore (2004) have been used successfully with concept landscapes.

All of these algorithms are based on the premise of much larger and denser graphs, therefore the results for concept landscapes of typical sizes do not differ very much (if at all) between the algorithms and running time can be neglected completely. The second case study uses community detection as part of the analysis. Also, the layout described below relies on communities.

### 7.2.3.4 Frequent Subgraph Analysis

Induced subgraphs can be used in the context of concept landscapes, for example, to reduce concept maps to a common set of concepts before aggregating them, or by reducing a landscape to a small number of concepts that one is interested in for analysis, as shown in the first case study.

However, subgraphs can also be the focus of interest themselves concerning, for example, the distribution and frequency of occurrence of the different structural combinations that a (small) number of concepts can have in a set of maps. It is unlikely that all structural configurations of, for example, four different concepts are appearing in equal numbers. Instead, there will probably be only a subset of the (exponentially) many configurations appearing at all. The distribution might show a clear preference for one or two structural configurations. This, in turn, may reveal something about the way these concepts is typically connected in memory.

When restricting analysis to subgraphs up to a certain (small) size and using efficient methods, concept landscapes can be analyzed in this vein. Hubwieser & Mühlung (2011b) present an algorithm developed in the course of this work that counts the occurrences of all structurally different subgraphs up to a given number of nodes. A similar approach is suggested by Yoo & Cho (2012). Grundspenkis & Strautmane (2010) suggest using graph similarity for small patterns in the context of automatically scoring concept maps.

For concept landscapes the subgraphs are typically undirected and the nodes are distinguishable entities: It is a difference whether concept  $A$  is connected to concept  $B$  or concept  $C$ , in general. The investigation of subgraphs gets computationally expensive rather quickly nonetheless. The different structural configurations are growing exponentially with graph size, making interpretation difficult. The software package CoMaTo presented in the next chapter therefore restricts the analysis of subgraphs to those of three or four concepts. For two concepts, the task resolves to finding out whether or not the concepts are connected in the graph. If concepts connected to themselves are not allowed, there will be  $2^{\frac{n^2-n}{2}}$  different structural configurations: There are  $n^2 - n$  positions in the adjacency matrix without the main diagonal that can be filled with either 1 or 0, but the matrix must remain symmetrical. For five concepts, this already amounts to more than a 1000 different structures - which is clearly beyond a size that can be interpreted manually. Since fast algorithms usually are not treating nodes as distinguishable entities, CoMaTo uses an exhaustive search that first creates all possible subgraphs of a given size and set of concepts. Then, each concept map of an accumulated concept landscape is first restricted to the subgraph of the given concepts and then matched to one

of the possible subgraphs. For the amounts of data used in the case studies and subgraphs of at most four concepts, the running time doesn't impose problems on the analysis so there is no need for computationally faster approaches. An analysis of the frequency of subgraph structures is presented in the first case study. It is a convenient way of analyzing the different structural combinations of a small subset of concepts in more detail. In theory, subgraphs can also be investigated independently of specific concepts. The result would then show whether or not certain structural arrangements are preferred per se.

## 7.2.4 Visualization

As concept landscapes are mostly a tool for data analysis, their visualization in this thesis deals with the task of visualizing the information that is used for analysis and contained within the landscape. As Chen (2002, p. 1) puts it:

“[I]nformation visualization can be broadly defined as a computer-aided process that aims to reveal insights into an abstract phenomenon by transforming abstract data into visual-spatial forms. The intention of information visualization is to optimize the use of our perceptual and visual-thinking ability in dealing with phenomena that might not readily lend themselves to visual-spatial representations.[...] [It] traditionally focuses on finding meaningful and intuitive ways to represent non-spatial and non-numerical information to people.”

This thesis only deals with the visualization of graphs, which has been proven useful in analysis. The visualization of, for example, vertical accumulations in form of a binary matrix is an interesting topic by itself, though.

In a way, a concept map itself is visualized information of the abstract phenomenon conceptual knowledge - which is especially obvious when using it as a teaching aid - this has also aptly been called “knowledge visualization” (cf. Cañas et al. 2005, p. 205). While a concept map usually already has a layout determined by the creator during drawing, a concept landscape doesn't as it is aggregated from many different maps. Also, aggregations often yield much larger graphs that cannot be easily grasped visually without a good layout. The task of visualizing amalgamated concept landscapes therefore amounts to the visualization of graphs which may be weighted, labeled, or show a temporal development.

Clearly, there are many available methods for visualizing graphs that can be used or adapted for the task at hand. Visualizing graphs is a broad task that can focus on many different, sometimes even mutually exclusive aspects. For concept landscapes the task focuses on the structure of a graph that is usually moderate in size. However, the structural information is only useful when combined with at least the labels of the nodes - the concepts. Part of the structural information is often contained in (usually numeric) attributes of the edges, like weights or scores. So, in contrast to the visualization of “big data”, like the traffic of the major internet backbones, for example, the focus usually does not lie on the “big picture” but more on a number of details that must be visible.

### 7.2.4.1 Vertical Landscapes

Visualizing vertical landscapes is equivalent to finding a “good” node layout for a graph and deciding if and how additional information like the weight of edges should be visualized. Visualizing edge weights is easily done by using differing line widths, as in e.g. (Eckert 2000). Other measures, like the centrality of a node, are sometimes displayed by differing nodes sizes, as in e.g. (Diethelm, Hubwieser & Klaus 2012). Glöggler (1997, p. 164ff.) describes a visualization of a set of graphical networks that are similar to concept maps that uses different sizes for nodes depending on their frequency of occurrence in the set.

There are many algorithmic solutions for automatically determining “good” layouts. A good layout is dependent on the context of its use. It may be “good” to avoid the crossing of edges or it may be “good” to find a visually appealing layout or even to keep the layout constant between different visualizations (e.g. a circle layout). A good layout for a concept map will present structural information to the reader while remaining a nice visual appearance and good readability.

For the figures in the next part, mostly a layout based on force was used<sup>2</sup> with some manual adaptations. A force directed layout calculates node placement based on a physical simulation of attracting and repelling forces between the nodes based on the structure (cf. Kaufmann & Wagner 2001, p. 71ff.). A similar method is used for analyzing single concept maps by Koponen & Pehkonen (2010).

Additionally, the following algorithmic approach has been successfully applied to concept landscapes, as presented in (Hubwieser & Mühling 2011a) and is also implemented in the CoMaTo package described in the next chapter:

---

<sup>2</sup>“Organic layout” as implemented in yEd, <http://www.yworks.com/yed>

1. Calculate a set of communities in the graph using a suitable algorithm.
2. Allocate space for each community around a circle.
3. For each community: Place all nodes of the community along a circle, which is then sized appropriately and placed in the allocated space of step 2.
4. Optimize the position of each node and of each community by using a suitable optimization algorithm.

Tests have shown, that using the Spinglass algorithm in step 2 and “Simulated Annealing” (Kirkpatrick, Gelatt & Vecchi 1983) in step 4 provides good results. Optimization is done by switching the positions of nodes within each community as well as switching the positions of whole communities (at random) and using the total length of edges on the Euclidean plane as a measure of quality. With increasing runtime (i.e. lower “temperature”) a change is more and more probably rejected if it doesn’t increase the quality.

Leaving out the last step results in a graph that has basically the same structure (many small circles of concepts arranged around a larger circle of communities), but will often be hard to read as there are many unnecessary intersections of edges and long edges that are connecting communities placed far apart. Fig. 19 shows the result of applying this method to a larger concept landscape taken from the first case study.

### 7.2.4.2 Horizontal Landscapes

For horizontal landscapes, the major point of interest usually is the development of a set of concept maps over time. Therefore, the visualization should focus on this aspect. In the course of this work horizontal landscapes were only used in order to display the development of graphs; either as a horizontal accumulation of a set of maps, or as the horizontal accumulation of a set of vertical amalgamations. In both cases, the data consists of a “time series” of maps that should be visualized.

One way of achieving this is by computing a “master” layout for the concept landscape that determines a fixed position for each occurring concept and then using this layout consistently throughout the visualization of the constituent maps. This has been previously suggested for the comparison of different concept maps (cf. Koponen & Pehkonen 2010, p. 1659). When visualizing the development of a concept map, it makes sense to use the layout of the latest map, as far as applicable. Some concepts will not appear in the last map and are therefore not covered by

this layout. These can then be placed manually, for example by placing them on a circle around the layout of the final map. CoMapEd, presented in the next chapter, is using this scheme in order to visualize the creation process of a user's concept map by horizontally accumulating all of the intermediate steps as separate maps. Also, in the first case study in chapter 10 there is an application of this scheme (Fig. 29 and Fig. 30).

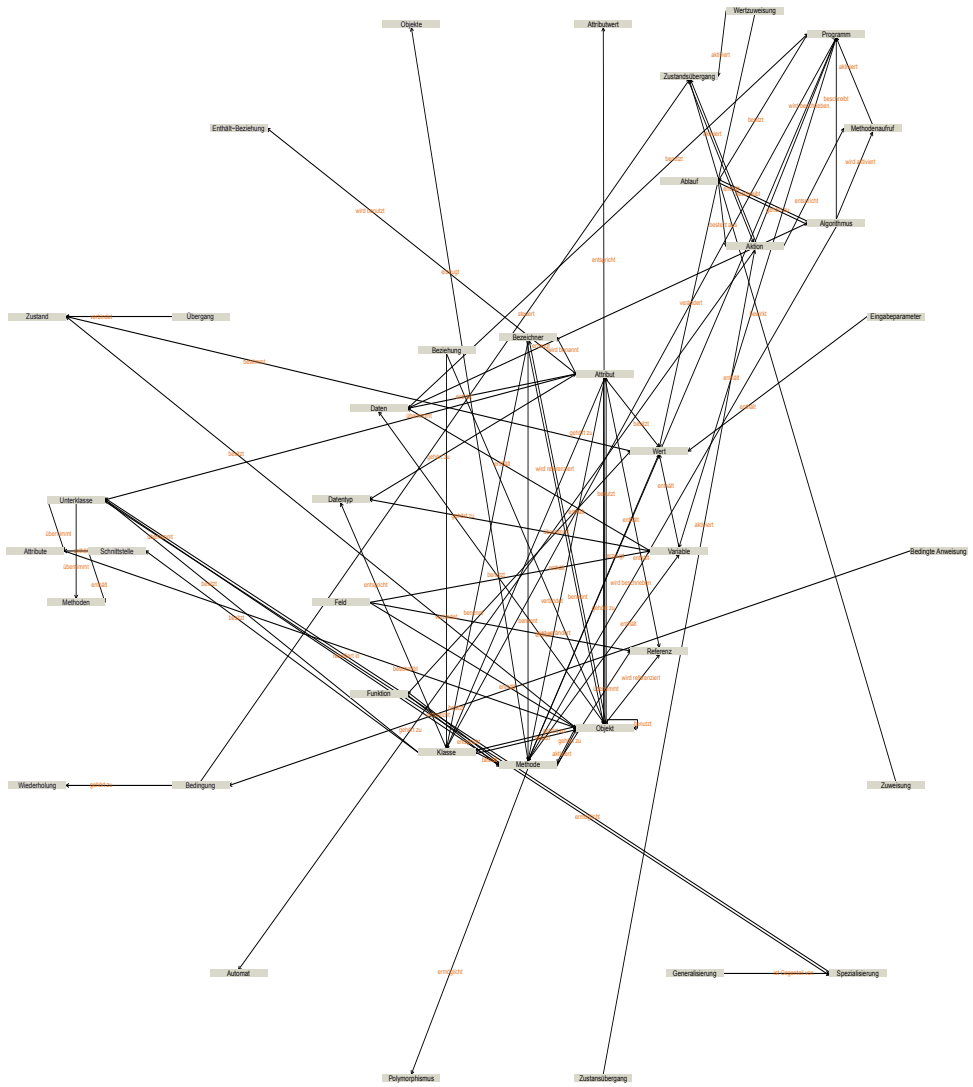


Fig. 19: The result of applying the layout algorithm to a concept landscape taken from the first case study. It is based on communities that are aligned on a circle, the largest community forms the center. Even though the node placement has been optimized, there are several improvements that could be made.





## 8 | Software Support for Concept Landscapes

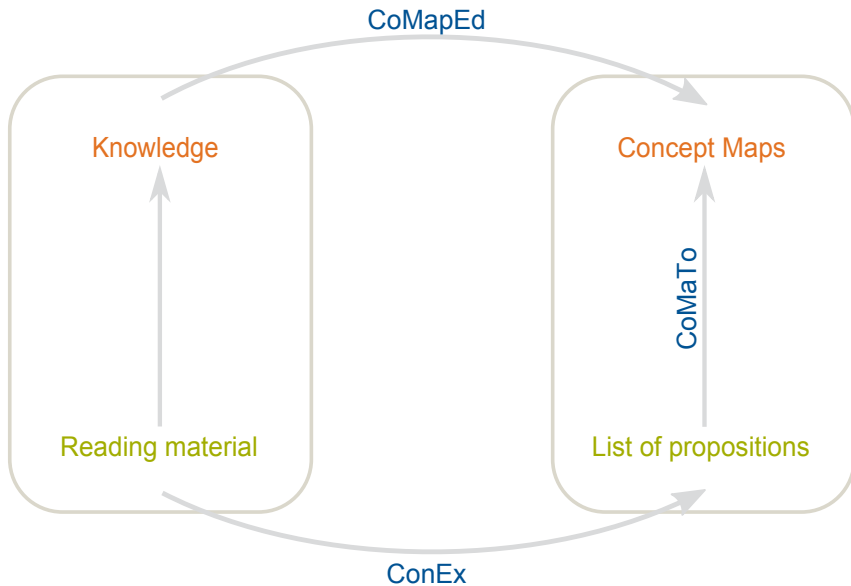


Fig. 20: The three software projects that support the pivotal points of the schema presented in Fig. 1

Following the central theme of this thesis - new ways of analyzing aggregated concept map data - the usefulness for computer support is self-evident: The more data there is, the more interesting the results will be, in general. But the more data, the more it becomes tedious or virtually impossible to collect, aggregate, and analyze it manually. This has also become obvious in the course of the studies that form the basis of this work, where some parts of the work have only been manageable by employing several students to help with the amounts of data. Therefore, if possible, software should be used in pivotal points to increase the amount of data that can be handled. Referring to the general research setting of Fig. 1, there are three processes that can be supported by software: The externalization of knowledge, the analysis of concept landscapes the analysis of (textual) input. Consequently, three software projects have been developed in the course of this thesis. Fig. 20 shows how they support the workflow. The next sections present these projects. Since software engineering itself is not the focus of this work, only a short overview over available software and their shortcomings, a list of requirements, and some notes regarding the implementation and final software are given.

## 8.1 CoMapEd

The collection of concept maps can happen in many ways. However, if the subsequent analysis is done digitally, the maps must either be collected electronically in the first place or digitalized later on. Obviously, the way of collecting the maps should not impede the externalization of knowledge itself. As has been noted in section 6.1.1, electronic solutions for creating concept maps are already in use and it seems safe to assume that there is no negative impact of using a software-based approach to concept mapping. Even more, it can be assumed that there is a positive impact from doing so. Another effect that is in favor of a software-based drawing is that the specific graphical syntax of concept maps can be “enforced” much more easily. Pen and paper based maps, as the experience of the case studies have shown, are often highly varying in the syntax in ways not consistent with concept maps, like using arrows that are connecting multiple concepts together. This section presents the software tool *CoMapEd (Concept Map Editor)*, which was designed and implemented for this purpose.

There are existing software solutions for drawing concept maps in various settings. Most notably, there is *Cmap Tools*<sup>1</sup> as described by Cañas, Hill, Carff, Suri, Lott & Gómez (2004). Most solutions are either not exclusively for concept mapping but more general drawing tools or are implementing enhanced versions of concept maps that go beyond the basic structure, like *COMPASS*<sup>2</sup>, described by Gouli, Gogoulou, Papanikolaou & Grigoriadou (2005a). Also, all of the solutions seem to be based on a program running on the user’s computer.

### 8.1.1 Requirements Analysis

A typical use of the system encompasses two kinds of users: *Participants* use the system to draw a concept map. *Researchers* use the system to collect the concept maps of participants in order to analyze them.

For participants, the system should present itself as an easy to use editor that allows them to draw concept maps in their basic form - labeled concepts that are connected by labeled propositions. Also, the editor should offer some additional benefits for the participants in order to keep their motivation for externalizing their knowledge as high as possible. A negative impact on motivation will most probably also have a negative impact on the complexity of the results. The additional benefits

---

<sup>1</sup><http://cmap.ihmc.us>

<sup>2</sup><http://hermes.di.uoa.gr/compasseng.htm>

include for example the possibility to save the map and continue to work on it later on or the possibility to export the map in order to print it, or continue to work on it with other programs on their own. The operation of the system must be mostly self-explanatory so that none or only very minimal prior training is required for the participants - assuming the participant has at least some experience as a computer user. Also, it should run on as many computer platforms as possible, with as few additional requirements as possible, so that the participants can use it at home or at any other place.

A researcher using the software should be able to manage typical research scenarios which include longitudinal and latitudinal surveys. Also, it must be easy to give access to participants. It should be possible to restrict the actual concept mapping task in the typical ways described in chapter 6 - restricting the list of concepts and restricting the list of edge labels that may be used. The collected data should then be exportable by the system and basic analysis should be provided directly within the system. Also, the process of concept map creation should be observable to the researchers.

The following is a summary of the functional (FR) and non-functional (NFR) requirements derived from these considerations:

- FR1** A participant must be able to draw a concept map consisting of graphical symbols with labels and labeled connections between those symbols.
- FR2** It must be possible for a participant to save and later load a map in order to continue working on it.
- FR3** It must be possible for a participant to export the map in formats that can be used by other programs.
- FR4** A researcher must be able to create and manage surveys for longitudinal as well as latitudinal studies.
- FR5** A researcher must be able to provide and restrict access to a survey.
- FR6** It must be possible to restrict the process of concept map creation to a certain set of concepts or to a certain set of possible labels for the connections between concepts. Also it must be possible to display a focus question or task description and to give a concept map as starting point for the participants.
- FR7** The concept maps belonging to a survey must be accessible/exportable by the researcher.

**FR8** Basic analysis of the concept maps of a survey should be supported within the system for the researcher.

**NFR1** The software must be easily operable for the participant, i.e. a person with typical experience in using computer programs should be able to draw a concept map with the software without prior training.

**NFR2** The software must have only minimal system-requirements in order to be accessible to a wide variety of participants.

**NFR3** The concept maps of the participants must be stored centrally, so that they are automatically accessible to the researcher, without the participant taking special action.

**NFR3** The system must record a history of each concept map.

**NFR4** A larger-scale survey with reasonably sized concept maps may not provide any problems regarding performance of the system.

### 8.1.2 Design and Implementation

The design of CoMapEd is based on several decisions that were made according to the requirements. In particular:

- To fulfill NFR2 and NFR3 from the list of requirements, a web-based system that runs in a browser is an obvious choice. Alternatives, like a client/server system would always require a program to be downloaded and run which in turn would also require a browser. This would have more requirements than a simple web-based system. On the downside, drawing concept maps within a browser requires modern technology, which might affect NFR2 in a negative way. Therefore, care was taken to keep the actual requirements for the browser as lean as possible. In the case of CoMapEd, this means that an approach using HTML5 and Javascript was chosen, instead of relying on different technologies like Adobe Flash.
- The software consists of a front end, that the participants use and a back end for the researchers. Concept maps are organized in surveys, as required. Surveys in turn are organized into projects, as this makes managing a longitudinal survey with several assessments easier.

- Concept maps are identified by a code (“slug”), that the participants can use to reload a map at any time. This also serves as a possibility for researchers to invite participants individually. Additionally, surveys can be made accessible by a code, which serves as a way of creating new maps anonymously. Each map will still have a personal code though, but the researcher do not know the connection of these to the persons, in general.

The design was implemented as a website using the *Ruby on Rails*<sup>3</sup> Framework. Additionally, for a modern and clean appearance, the *Bootstrap*<sup>4</sup> CSS-Framework was used. The drawing of the concept maps is done with the SVG support built into HTML5 and is based on the *D3*<sup>5</sup> (*Data-Driven-Documents*) Javascript library.

Fig. 21(a) shows the main screen that a participant is using to create a map. The drawing area forms the largest part of the screen. All actions are available in the menu on the left hand side of the screen. The drawing itself can be done either from the menu or by using the mouse on the drawing area. Additional information, if necessary, is presented in a box at the top. Directly after logging in an introductory text is presented there, which can be defined by the researcher and can include the focus question. Additionally, for example, online help/instructions can be shown there if the user selects this action from the menu. Also, if a concept map is opened for the very first time, a short introductory text about concept mapping including an example is presented to the user before the main screen appears. Currently, the map can be exported in two formats: As vector graphic (SVG) or in the simple graph format (TGF) that can, for example, be opened by yEd.

Fig. 21(b) shows the back end view that is presented to the researchers. The projects and surveys are organized in the left vertical menus. The concept maps of a survey are then presented horizontally, including a preview of the map, basic information and the possibility to view the history of its creation. For this, the layout described in section 7.2.4.2 for horizontal landscapes is used. Maps, surveys, and projects can be exported and imported. Additionally, a “filter” view allows creating and exporting an amalgamated landscape from the set of all concept maps present in the projects of a researcher.

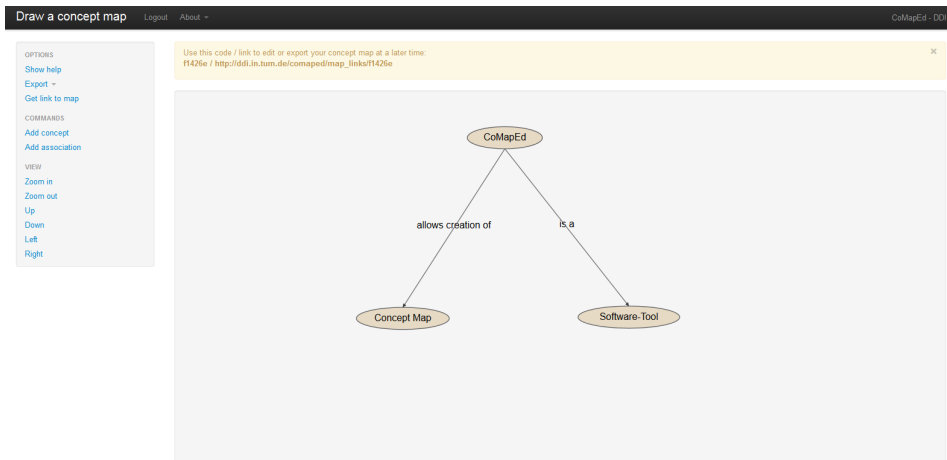
The system has been used to collect almost a 1000 maps in several settings, so far. Both front end and back end are available in English and German, depending on which language the browser requests (English being the default). The feedback of users has been generally positive.

---

<sup>3</sup><http://rubyonrails.org>

<sup>4</sup><http://getbootstrap.com>

<sup>5</sup><http://d3js.org>



(a) Frontend

The screenshot shows the CoMapEd backend interface. At the top, there is a header with "Backend" and "CoMapEd - DDI - Backend". Below the header is a navigation menu with options like "Projects", "Filter", "Users", and "Logout". The main area displays a survey overview for "Längsschnitt WiSe 13/14". The survey is available, with a date of 2013-10-23 and a code of infgwis13. The survey is open, with a date of 2013-11-07 19:32:57 UTC. The survey has 4 accesses, 78 versions, 33 concepts, and 36 associations. The survey is titled "Längsschnitt WiSe 13/14" and is currently viewed at page 741. The survey is titled "Längsschnitt WiSe 13/14" and is currently viewed at page 741. The survey is titled "Längsschnitt WiSe 13/14" and is currently viewed at page 741.

(b) Backend

Fig. 21: Screenshots of the front- and back end of CoMapEd.

## 8.2 CoMaTo

The analysis methods presented in the last chapter have all been presented with the purpose of implementing them in software. This section presents the software package *CoMaTo* (*Concept Mapping Tools*) that allows working with digitally stored concept maps in GNU R.

Cañas, Bunch & Reiska (2010) present analysis software that is extensible and is built upon the data format of Cmap Tools. It may be possible to employ it for use with concept landscapes. However, especially when considering that an analysis of concept landscapes can include many different aspects of statistics and can often be exploratory in nature, using a platform like R for the task provides many benefits. There are ways of using R together with existing packages like *igraph* to analyze concept landscapes directly. Nevertheless it is worthwhile to implement the methods in an explicit package for reuse and for ease of analysis. Consequently, all analyses presented in the next part have been done with CoMaTo.

### 8.2.1 Requirements Analysis

The central application of CoMaTo is the computer aided analysis of concept landscapes using the methods presented in the last chapter. The following list of requirements has been identified:

- FR1** The package must be able to import concept map data in different formats. Especially, it must be able to import the export of CoMapEd.
- FR2** The analysis techniques as presented in section 7.2, if applicable, must be supported by the package.
- FR3** The package must provide data structures that allow applying other analysis techniques as well.
- FR4** It must be possible to display or export the results (e.g. for preparing scientific publications).
- NFR1** The software must be easy to use for a person already experienced with GNU R and its package system.

**NFR2** A typical survey with reasonably sized concept maps must be analyzable without posing problems regarding memory usage and running time to a current computer.

## 8.2.2 Design and Implementation

The package is based on the “S3” object orientation of the R programming language and offers two basic data structures/classes: `concept.map` embodies a single map and `concept.maps` a set of maps. Each concept map is represented as a graph that is stored in an `igraph` object. Concerning the input capabilities, the differing formats of data that appeared in the various case studies have been incorporated. This includes concept maps stored in spreadsheets as list of propositions as well as the two graph formats *TGF* and *GraphML*. Also, the system can be used flexibly as there also exists a constructor that simply accepts a R matrix as input. Concerning the output, the package can generate R plots and TGF files, e.g. for pathfinder networks.

Based on the two data structures, a set of functions is available to work with the data. Most notably, the function `landscape` implements the aggregation of a `concept.maps` object. Vertical amalgamations and accumulations are supported in an extensible manner. The creation of Pathfinder networks from a set of concept maps and the clustering using multivariate Bernoulli mixtures as described in the last chapter are implemented as analysis methods. Both basic and advanced graph measures are also available - mostly based on the corresponding implementations of the `igraph` package. Visualizations and horizontal landscapes are supported in form of the possibility of plotting both a single map and a series of maps using the layout algorithm described in section 7.2.4.

## 8.3 ConEx

This section presents the software *ConEx (Concept Extraction)* that allows the automated analysis of texts. While not a central aspect of this thesis, the analyses of texts that are used as input for learning is nevertheless a worthwhile endeavor. The goal is, to extract salient concepts and propositions that are based on these concepts from text data automatically. While a manual analysis is possible and has been done for all of the case studies of the next part, an automated analysis is far easier, better reproducible, less prone to spurious errors and allows the processing



of much larger texts. This becomes important especially for content domains where experts don't necessarily agree on the structure and importance of concepts (cf. Trumpower et al. 2010, p. 24) and therefore querying an expert might not yield the desired results.

There are many existing suggestions and implementations of programs that work with texts, also for the specific task of extracting salient terms from a corpus. In the course of the research projects, the systems described by Eynard, Marfia & Matteucci (2010) and Schöneberg (2010) have been investigated, also *MaxQDA*<sup>6</sup> has been used to identify nouns in texts together with their absolute frequency and also to identify the position where these appear in a text corpus.

### 8.3.1 Requirements Analysis

The following list of requirements has been identified:

**FR1** The software must work with texts in English and should also work with texts in German.

**FR2** The software must be able to extract salient concepts (according to some measure) and a list of sentences in which these concepts appear from text stored in a file.

**FR3** The user must be able to modify the list of salient concepts prior to the extraction of associations.

**FR4** The results must be exportable in a suitable format.

**NFR1** The software must be extensible concerning languages and measures of salience.

**NFR1** The software should offer support for different, common file format regarding its input.

### 8.3.2 Design and Implementation

ConEx has been implemented as a stand alone Java program. It does not present a novel approach to text analysis, but instead is a software solution specifically

---

<sup>6</sup><http://www.maxqda.com>

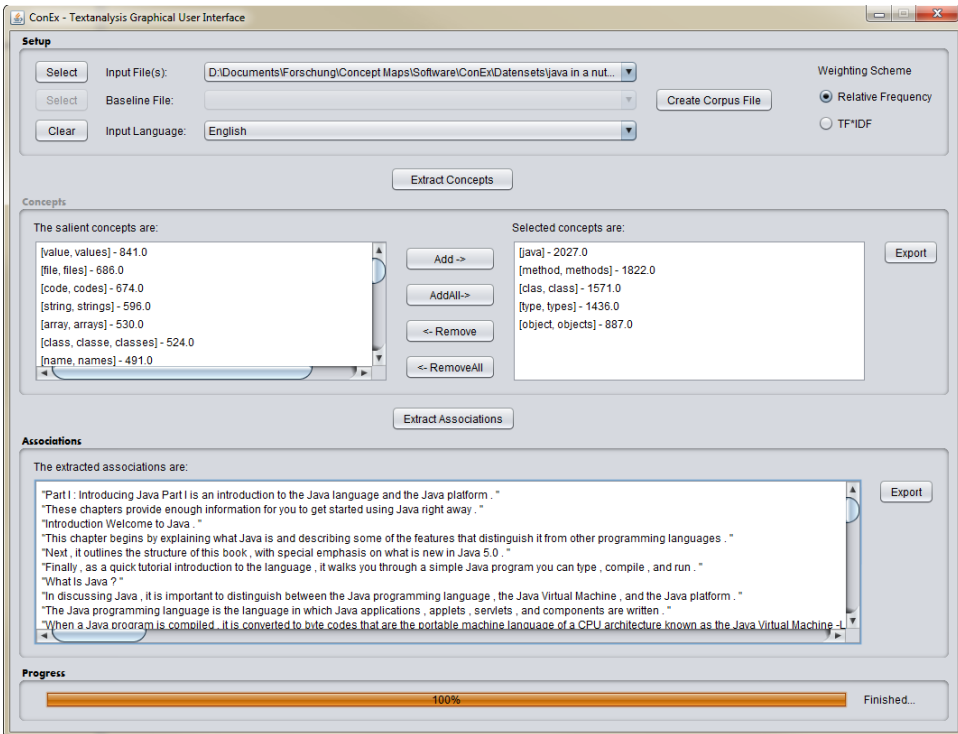


Fig. 22: Screenshot of the graphical user interface of ConEx.

designed for the task at hand that builds upon established methods and existing software. It is based on the NLP toolkit and WordNet, presented in section 5.4.1.

Fig. 22 shows the graphical user interface<sup>7</sup>. A version that can be operated from the command line is also implemented. The user interface follows, from top to bottom, the basic workflow of choosing input files and general settings, then extracting concepts, and finally extracting propositions (associations, or sentence) from a subset of the extracted concepts. Both PDF and plain text files are supported as input. Currently, English and German are supported as input languages ( WordNet is only available for English). As measures of salience both the absolute frequency and the TF\*IDF weighting scheme (see section 5.4) can be used. In the latter case, a corpus of texts has to be given which defines the “baseline” frequency of the words in the weighting. Both the concepts and the associations can then be exported in the CSV format. The object-oriented design of the program and a suitable class hierarchy allow the easy inclusion of new languages, file formats or the inclusion of translations of WordNet. The extraction algorithm for the concepts works like this:

<sup>7</sup>The user interface has been developed by Reinhard Hahn as part of a student project

1. Tokenize the input text.
2. Classify the tokens of each sentence.
3. For each noun:
  - (a) If available for the chosen language: Create a baseform of the noun using WordNet.
  - (b) Store the number of occurrences for the noun. Nouns are treated as equal if they have the same baseform (or by simple string comparison if no baseform is available).
4. Return the salience of each noun as defined by the chosen measure.

The associations are then extracted by using the tokenized sentences. Each sentence in which at least one of the selected concepts appears is added to the output. This may include sentences in which not every appearing concept is also on the list of selected concepts. Selecting only sentences in which every appearing concept is on the list of selected concepts will, in general, incur the loss of a lot of information. Therefore, subsequent analysis steps must take into account that not every sentence of the generated output may be viable for using it in the analysis of learners' knowledge artifacts.



## **Part IV**

---

### **Case Studies**



## 9 | Overview

This part presents the results of three case studies CS1 to CS3 that were originally conducted as research projects and led to the development of concept landscapes. As part of this thesis, the analyses were redone using the new notions. Each chapter describes one of the studies and they are all structured alike, reminiscent of a research publication.

First, the general context and specific setting of the study is presented in detail. References to related research and literature are also included, where appropriate. The choice to exclude these references from the related work part was made intentionally as the settings of the case studies are not directly relevant for any other parts of this thesis. Next, the collected data is presented in detail together with the research question that the experiment focuses on. The data always is centered around concept maps, usually accompanied by additional information that was collected or otherwise available for analysis. The specifics of the concept mapping task are described there as well. Then, the analysis methods and results are presented. Care is taken to clearly state the aggregation method and the analysis technique. The presentation of the results is separated from their discussion, which forms the final section. There, the answers to the research questions are given, as far as the results allow it.

Two of the studies - including the experimental results - have been previously published in the course of this thesis. The relevant publications are cited at the beginning of each chapter. Also, some text parts of the following chapters have been taken in verbatim from these publications but are not shown as quotes here, for better readability. All of the analysis results and figures were re-done for this thesis - a close similarity to the respective publications is necessarily still present, though.

Taken together, the case studies serve to give an overview over the typical workflow of an analysis using concept landscapes as presented in section 7.1 and defined in section 7.1.1 as well as the analysis methods presented in section 7.2. The strengths and weaknesses of the techniques as well as the typical way of interpreting the results will be shown. As has been mentioned in chapter 2 already, the case studies all are research projects in their own merit. So, the results are also providing valuable insight into the fundamentals of computer science education and are more than mere test beds for the new analysis methods. The following description gives a quick overview over the major facts of each project:

**CS1** Computer science education for non-majors

**Focus** Monitoring the knowledge development in the course of an introductory CS lecture to non-major students in their first semester.

**Results** Despite the difficult setting concerning the students' motivation, they don't necessarily resort to rote-learning and the learning process according to the models of learning presented in chapter 3 can be made visible.

**Conducted in** October 2010 to February 2011

**Collected concept maps** : 90

**Participants** Non-major students of an introductory lecture

**Analysis methods** Basic graph measures, subgraph frequencies, horizontal visualization

**Results previously published** Hubwieser & Mühling (2011a) and Hubwieser & Mühling (2011c)

**CS2** Knowledge structures of beginning CS students

**Focus** Beginning CS students and their conceptual knowledge about core CS concepts. Also, measuring the impact of a newly introduced compulsory school subject "Informatics".

**Results** The knowledge structure of a beginning CS student is rather complex. In general, students who attended the school subject have a different knowledge structure than the others, showing artifacts of a more formal CS education.

**Conducted in** April 2011 and October 2011

**Collected concept maps** : 590

**Participants** Beginning major students

**Analysis methods** Pathfinder networks, advanced graph measures, community detection, cluster analysis

**Results previously published** -

**CS3** Conceptual knowledge and abilities

**Focus** Development of the knowledge of concepts related to object-oriented programming in a preparatory course for beginning CS students. Only minimal teaching input was given - instead self-guided learning-by-doing was stressed. Additionally, the source code was analyzed in order to relate the conceptual knowledge to programming abilities.



**Results** It is possible to have students develop their mental models with only very little input but practical programming experience. Also, the knowledge about concepts and the ability to use them successfully is interrelated in different ways.

**Conducted in** October 2010

**Collected concept maps** : 188

**Participants** Beginning major students

**Analysis methods** Pathfinder networks, cluster analysis

**Results previously published** Berges et al. (2012)



# 10 | CS1: Computer Science Education for Non-Majors

Most parts of the results - in particular the description of the context and the collected data - have been previously published in (Hubwieser & Mühling 2011c). Additionally, some results have been presented in (Hubwieser & Mühling 2011a).

Today's world is heavily dependent on computing technologies and electronics. Consequently, the demand for computer science education in nearly every institution has risen (cf. Guzdial 2003). Especially at universities, fields of study with a more technical background will typically have some form of computer science education in their curriculum. For example, almost every major of engineering at the TU München has compulsory CS education in the first semesters. Sometimes, the courses are specialized for the particular field of study (e.g. focus primarily on writing short scripts in Matlab), or they provide a general overview over the basics of programming and computer science.

Teaching computer science to non-majors is a challenge: "Studies of the problem point to the overemphasis in computer science classes on abstraction over application, technical details instead of usability, and the stereotypical view of programmers as loners lacking creativity" (Forte & Guzdial 2004, p. 1). Also, the motivation of students in a non-major course is usually different from those in major courses, as typically, their main area of interest is not computer science but some other field. "Historians, writers, architects, and engineers (just to name a few) have diverse interests and require different kinds of computational proficiency to perform the tasks that are important to them" (Forte & Guzdial 2004, p. 3). Therefore, and given that typically the prior knowledge of CS related concepts will be only weakly developed, it is much more likely that non-major students will resort to rote-learning in computer science classes, unless they see a clear relevance, are motivated, and decide to learn meaningfully. Investigating the knowledge structures of non-majors can reveal whether or not this actually happens.

## 10.1 Description of the Setting

This case study presents the results of investigating a lecture that is held over the first two semesters for students with a major of geodesy at the TU München. The lecture has been held by the working group "Didactics of Informatics" as part of

a more than 10-year cooperation. This chapter presents results of a study that was done in the winter term of 2010/2011, where the course was held as a typical combination of lecture and problem session. In the following winter term it was completely restructured, described in more detail in German by Berges et al. (2013).

The course was designed as a one semester lecture dealing mostly with object-oriented programming, presented in an objects-first approach, closely following the 10th grade of the subject Informatics in Bavarian secondary schools presented in the next case study. In detail, the course contents were:

**Chapter 1: Modeling** Informatics: main subject areas, typical working methods; Functional modeling: data flow diagrams; modeling techniques in computer science.

**Chapter 2: Object Oriented Modeling** Objects in documents: object, class, attribute, method, class card, object card; artificial languages: grammars, BNF; states of objects: state, transition, state diagram, real and program objects; object diagram, association, class diagram, multiplicity of associations, compound objects: creation of objects as values of attributes.

**Chapter 3: Algorithms** The concept of algorithms: programming languages, class definition: definition and declaration, signature of methods, access modifier, attribute declaration, definition of methods; structure of algorithms: graphical representation of algorithms, structural components of algorithms, nesting of components, input and output of algorithms; properties of algorithms: terminating, deterministic, determined.

**Chapter 4: Object Oriented Programming** Definition of classes: structure of object-oriented programs, definition and declaration, signature of methods, access modifier, attribute declaration, definition of methods; assignment statement, ring exchange, assignment in constructor methods, encapsulation, equality; translation of computer programs, compiler vs. interpreter, execution of programs, course of events of a program; communication by methods: input, output, side effects, local and global variables/attributes; creating objects at runtime, constructor method, references, removal of objects; implementation of algorithms: structure elements in programming languages: sequence, conditional statement, repetition; arrays, index.

**Chapter 5: State Modeling** Finite automatons, triggering and triggered action, state chart; Implementation of automatons: switch statement; conditional transitions: complete state modeling, implementation of conditional transitions.

**Chapter 6: Interaction and Recursion** Implementation of associations: unidirectional, bidirectional, 1:1, 1:n, m:n multiplicities, association class; sequence charts: calling of methods, sequence charts; Recursive algorithms: linear and cascading recursion.

**Chapter 7: Inheritance** Generalization: Sub- and super classes, specialization, inheritance; implementation of specialization, overriding of methods, generalization, class hierarchies; polymorphism: calling methods of foreign classes, abstract classes.

## 10.2 Data Collection & Research Questions

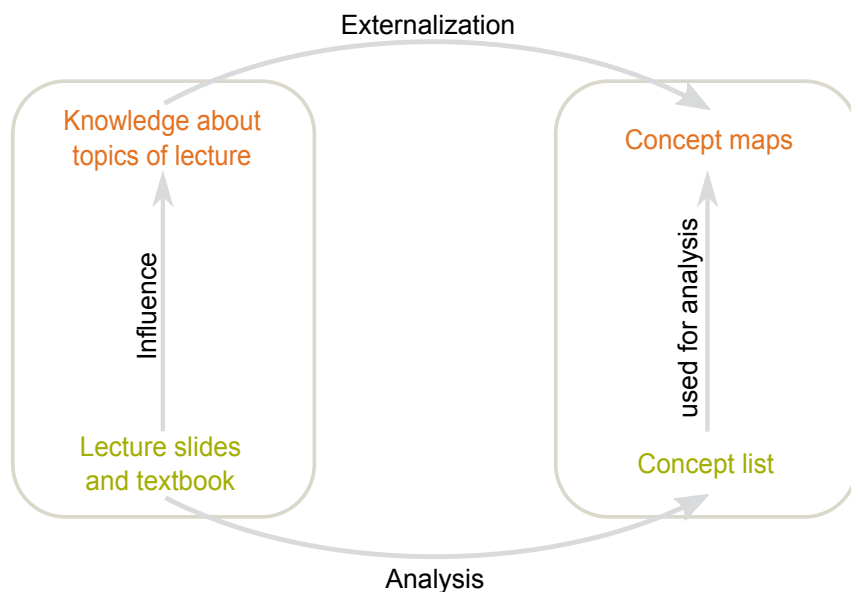


Fig. 23: Overview over the study, based on the schema of Fig. 1.

The students of the lecture were investigated closely in a longitudinal research setting in order to monitor the development of their knowledge. The students were repeatedly asked to draw concept maps anonymously. A code was given to the students that they should use instead of their names. In this way, the concept maps of each person could be matched without being able to match them to the actual person. Additionally, the students were asked to participate in a voluntary short exercise in the middle of the semester, again using their code. In this way, there was some way of relating conceptual knowledge to the results of applying this knowledge

in a short exam-like exercise. The students were asked to draw concept maps at four points in time over the course of the semester:

1. Right before the very first lecture (pre test, PRE),
2. after 4 weeks of lecture (mid test 1, MT1),
3. after 8 weeks of lecture (mid test 2, MT2), and
4. after 14 weeks, right after the last lecture (post test, POST).

The written exercise was one week before MT2, the final exam was after POST. The data of the study consists of the concept maps of the four points of measurement and the scores of the exercise.

Participation began with all 38 students of the course in the pre test. MT1 was attended by 33 students. For MT2 and POST, 19 students participated. The exercise was completed by 17 students.

Concerning the concept mapping task itself, the students were always given a (restricting) list of 40 concepts. They were told to use only concepts from this list and to use only those concepts that they were familiar with. The focus question was to try to draw a concept map from as many of the concepts they were familiar with as possible. The students did produce a new concept map each time. The first two tests were done using pen and paper and the second two tests were done using the graph drawing software yED. Fig. 24 shows one of the maps produced with yEd in the second midtest. The participants had roughly 30 minutes time for each concept mapping task. They received a short oral introduction to concept mapping at the first test. A written introduction was given to them each time.

To arrive at the list of concepts, the following semi-automatic procedure was used on the slides as well as on the main recommended textbook of the course:

1. Reduce the text material of the course to simple statements by removing all explanatory sentences, examples and sentences that don't carry relevant information. Remaining are sentences like: "If an attribute is marked as private, only objects of the same class are allowed to read or write its value."
2. Remove all non-nouns from the text and convert all nouns to singular and to the nominative case.
3. Sort these descending by their frequency of occurrence.

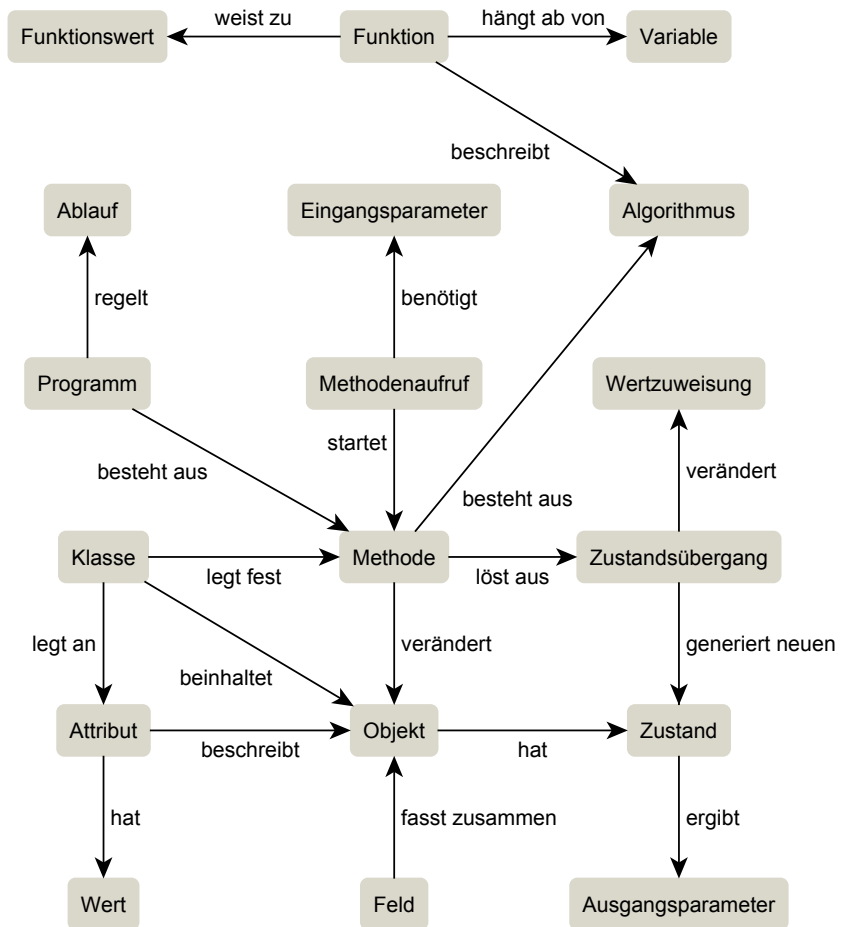


Fig. 24: An example map of MT2 as originally produced by one of the students in German.

4. Separate combinations of nouns that have their own meaning in the context (e.g. in German “Attributwert” is separated into “Attribut” and “Wert”)
5. Combine all nouns that have no meaning on their own in the context (e.g. “Garbage” and “Collection” is combined to “Garbage Collection”).
6. Remove nouns that are not part of the subject matter, that are too general (e.g. “Model”), too specific, technical (e.g. “RAM”), or proper nouns (e.g. “Pascal”).

While the process is not fully objective in the last step, it turned out to be rather objective and reproducible by letting a second person follow the same guidelines and comparing the results. The result was the following list of 40 concepts:

*action, aggregation, algorithm, array, assignment, association, attribute, class, condition, conditional statement, data encapsulation, data, data type, flow, function, function value, generalization, identifier, inheritance, initialization, input parameter, instantiation, interface, loop, method, method call, object, output parameter, polymorphism, program, reference, specialization, state, state machine, state transition, structural elements, subclass, transition, value, variable.*

ConEx, as described in chapter 8, was also used on the same (unprocessed) input data. When taking the first 40 concepts sorted decreasingly according to their frequency and manually filtering out different word forms (e.g. plural), there is an agreement of more than 60%. The following concepts are appearing in both lists, sorted by their frequency in descending order: *class, object, algorithm, reference, subclass, value, attribute, flow, program, identifier, loop, conditional statement, association, variable, function.*

Additionally, the following concepts are not appearing in verbatim, but have a close correspondence to the concepts given in parentheses: *call (method call), parameter* and *input* (both *input parameter*), *return value (output parameter)*, and *structure (structural elements)*.

Finally, these 13 concepts are not appearing in the manually derived list in any form (again sorted according to frequency): *java, set, count, case, order, type, superclass, definition, language, statement, implementation, programming language, page, sequence, description, constructor, relation, form.* Most of them would have been filtered out in the last step, which is missing in the fully automated concept extraction. All in all, using ConEx would have made the concept extraction easier; even without any manual intervention, the result would have an agreement of more than 60%.



Using these concepts, the text material was analyzed semi-automatically according to the following process:

1. Identify all propositions/sentences in the course material automatically by searching for occurrences of one or more of the extracted concepts.
2. As concept maps can only reflect monodic or dyadic associations, check the arity of the associations and keep only these.
3. Translate the information that is contained in the remaining sentences into propositions by qualitative means. If this is not possible unambiguously (e.g. due to a too complicated structure of a sentence), disregard the sentence.

Among the original 161 sentences, 101 contained two, 40 contained three, 17 contained four, and 3 contained five concepts. Therefore, most of the associations (63%) are dyadic and can be represented in a concept map. These sentences were then used as propositions for four “expert” concept maps (EM1 - EM4) containing the propositions and concepts that were presented to the students up to each point of measurement (including the exercise). When a pair of concepts was used in several propositions, only the first was added to the map. Therefore, the expert maps are mostly relevant concerning their structure. Fig. 25 shows the first expert map, EM1. The others can be found in the Appendix. Also Fig. 19 shows the structure of an amalgamation of the four maps EM1 to EM4 without regard to the temporal ordering.

Fig. 23 shows how this particular study can be represented in the basic scheme presented in the introduction of this thesis. The research questions are:

**RQ1** How does the knowledge of the students develop?

**RQ2** Which concepts are “common knowledge” among the students and which concepts provide the most difficulties to the students?

**RQ3** Can the investigation reveal insights about the process of learning?

## 10.3 Analysis and Results

The propositions of the concept maps were scored using the technique “relational scoring with master map” as described by McClure et al. (1999). The “master map” is the amalgamation of EM1 to EM4. The propositions of the student maps were

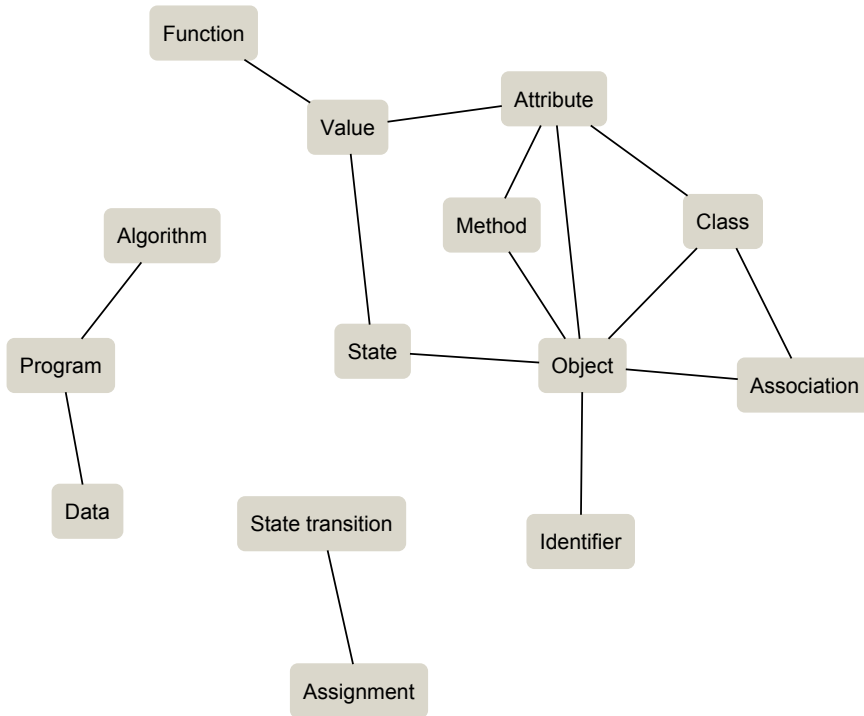


Fig. 25: The first expert map containing all the propositions that were presented up to MT1.

scored using the three values 0, 1 and 2, where 0 and 2 represent the cases of incorrect and correct propositions with regard to the master map. The value of 1 is for cases where neither 0 nor 2 can clearly be used, i.e. a “partially correct” proposition. Also, for some of the analyses presented below, transformed scores of -1, 0, and 1 instead of 0, 1, and 2 were used. Summing these score values gives subtly different results in the sense that only the extreme values are actually affecting the sum. Also, a negative and a positive score will cancel out in summation.

### 10.3.1 RQ1: Development of Knowledge

To analyze the knowledge development, the concept maps of each point of measurement were accumulated vertically by using four different mapping functions that map each concept map to its number of nodes, edges, components (i.e. ruggedness) and sum of edge scores respectively. The resulting vector. Fig. 26 shows the

r	Concepts	Propositions	Clusters	Score
Concepts	-	0.95**	0.39**	0.60**
Propositions		-	(0.18)	0.56**
Clusters			-	0.33**

Table 10.1: Spearman's rank correlation between the four basic graph measures. As score, the sum of all proposition scores of a map was used (\*\* =  $p < 0.01$ , () =  $p > 0.05$ ).

development of these basic graph measures over the course of the lecture when using mean and standard deviation of the vectors resulting from each accumulation.

It is interesting to see how closely the number of concepts and propositions follows each other. As the maximal number of edges for  $n$  nodes is  $\frac{n(n-1)}{2}$ , the number of concepts will always provide an upper bound on the number of propositions. However, the data shows that there are approximately only as many edges as there are nodes in the concept maps, as also noted in 6.1.1. Also, except for the edge scores, the variance increases over time. The increase in ruggedness (i.e. the number of components) and the stagnation of the edge scores is another interesting aspect.

To investigate the usefulness of simple graph measures further, the correlation between each pair of measures was calculated using all maps of all four points in time and calculating Spearman's rank correlation. This measures the degree of (linear) dependence between any of the two measures when only assuming they are an ordinal scale. The score of a map was calculated using the transformed scores and summing over all proposition scores of a map. Table 10.1 shows the results. Not surprisingly, the measure of vertices and edges correlates almost perfectly. For ruggedness there is no large effect visible when compared with any of the other properties, which is interesting to note. The sum of scores shows a medium to large effect to all of the other measures. All values are significant with  $p < 0.01$ , except for the correlation between the number of edges and the number of clusters.

Finally, the subgraph frequencies of the four concepts *attribute*, *object*, *state*, and *value* are investigated. The lecture material contains the (tetradic) proposition: "The state of an object is determined by the values of its attributes". As has been argued in section 6.2, it is not possible to encode such a complex proposition in a concept map unambiguously. The semi-automatic processing of the lecture material leads to a structural configuration of a circle (as also shown in Fig. 25), based on the four propositions:

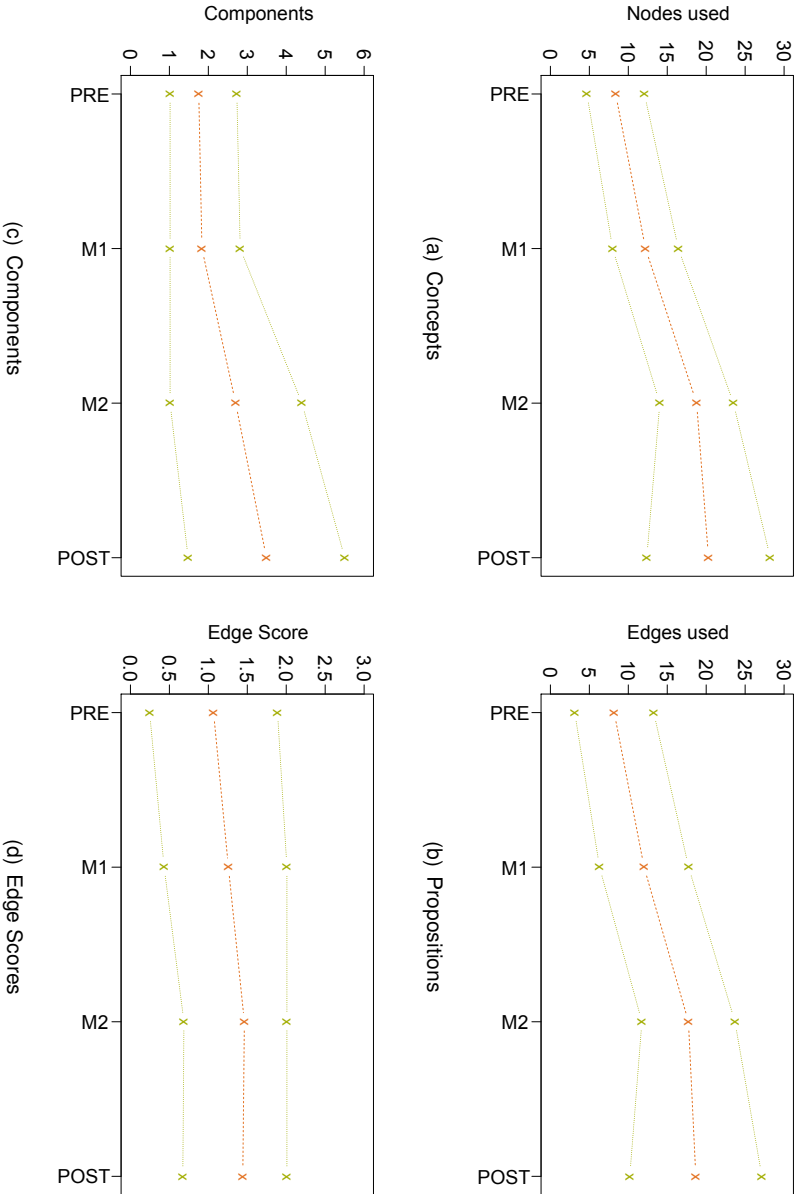


Fig. 26: Development of basic graph measures. Plotted is the mean of each measure over the four points of measurement. The top and bottom lines show the standard deviation from the mean. Where these would have exceeded the minimal or maximal actually occurring value, those are used instead, indicating an asymmetrical distribution.

1. *attribute* describes *object*.
2. *object* has *state*.
3. *value* determines *state*.
4. *attribute* has *value*.

Fig. 27 shows how the students chose to connect these four concepts. There are 64 different structural configurations, most of which have never been used. When ignoring the configurations appearing only once, five different subgraphs are occurring. The majority of maps are showing a completely empty subgraph (number 64 in the diagram), i.e. there aren't any connections between the four concepts. The other four are shown in Fig. 28. For MT2, only the empty subgraph and the ones with number 24 and 28 were used by students in more than one map.

When analyzing the subgraph frequencies of the three concepts *object*, *class*, and *attribute* - which form a fully connected triangle in the master map - the subgraph showing most frequently (12 out of 33 maps) is missing an edge between *class* and *attribute*. The two patterns appearing most frequently after that (at 6 maps each) are the triangle just like in the expert map and the empty graph.

### 10.3.2 RQ2: Common Knowledge and Misconceptions

To investigate the “difficulty” of concepts, only a subset of the list of concepts was used, since it is hard to visually keep an overview over the evolution of too many concepts. The lecture focuses mostly on object orientation. Therefore the analysis was restricted to concepts from this particular topic. A list of core concepts for this part of CS has been suggested as “quarks” by Armstrong (2006). There are nine verbatim correspondences of the quarks and the concepts in the list: *aggregation*, *association*, *attribute*, *class*, *inheritance*, *instantiation*, *method*, *object*, *polymorphism*. Additionally, the following six concepts of the list have a close correspondence to one or more of the quarks (given in parentheses): *data encapsulation* (*encapsulation*, *information hiding*), *data type* (*abstract data type*, *typing*), *generalization* (*abstraction*, *class hierarchy*), *method call* (*interaction*, *message passing*), *specialization* (*class hierarchy*, *reuse*), *subclass* (*class hierarchy*, *extensibility*, *relationship*, *reuse*).

To select concepts systematically and based on literature, these nine plus six concepts that are directly contained in or closely resembled by the list of quarks were

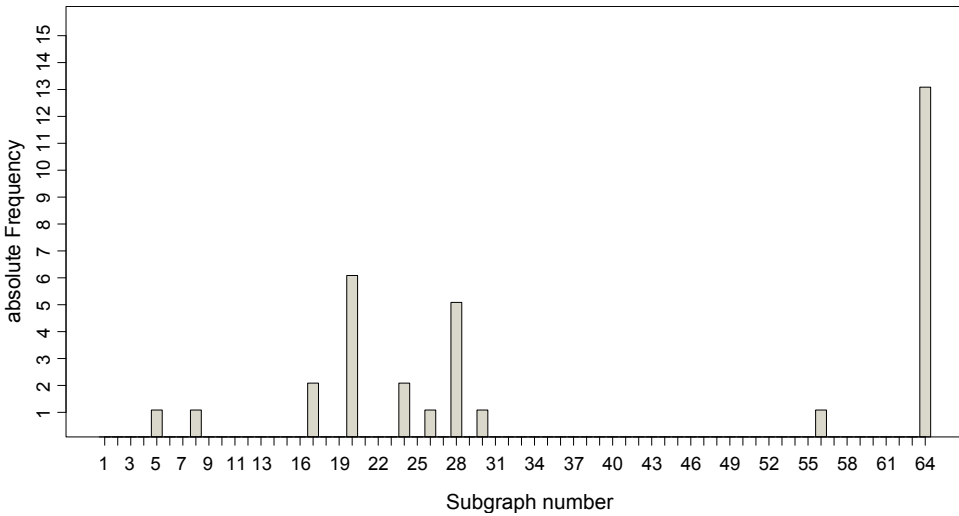


Fig. 27: The distribution of subgraphs of the four concepts *attribute*, *object*, *state* and *value* in the concept maps of MT1.

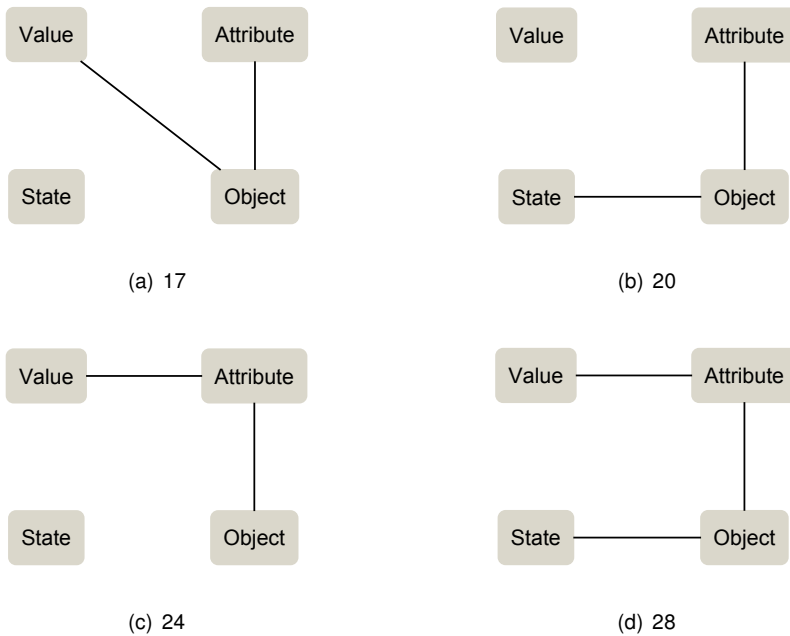


Fig. 28: The different subgraphs with more than one occurrence in MT1 - except for the empty subgraph. The numbers are in reference to Fig. 27

used. The list is thus: *aggregation, association, attribute, class, data encapsulation, data type, generalization, inheritance, instantiation, method, method call, object, polymorphism, specialization, subclass*.

The concept maps were restricted to these concepts and propositions between them. Then, they were amalgamated vertically, forming a weighted graph from the maps by summing the transformed edge scores (-1, 0, and 1). Since the analysis is focusing on “common knowledge” and “misconceptions” in the data, this new scoring scheme is more appropriate for the aggregation. The generated graph has  $15 \cdot 16/2 = 120$  undirected associations. Edges that represent associations that never appear in any of the maps receive a weight of 0.

The four resulting landscapes are then accumulated horizontally to include a temporal ordering. This accumulation was done in two ways. First, only edges with a “high” weight were kept and the rest were pruned, then only edges with a “low” weight were kept. “High” and “low” are defined as the 4th and 1st quartile of the range of edge weights, respectively. So, an edge with a “high” value means that at least 75% of the edges have a lower weight and conversely a “low” value means that 75% of the edges have a higher weight. Quartiles provide a relative measure. The “top” or “bottom” 25% of edges will always remain in the graph. A relative measure is useful, as otherwise the total number of maps would influence the result. Alternatively, an attribute like “proposition must have received a score of 1 in at least 50% of the maps” could be used. However, since it is unclear how such an attribute might look like, quartiles were used.

Fig. 29 and Fig. 30 show a visualization of the accumulated graphs over the four measurements for the high scoring and low scoring edges respectively. Again, a low scoring edge is taken as a misconception.

For every measurement of Fig. 29, the concepts *attribute, class, method, method call, and object* are present, even though the actual edges connecting them are varying. Nevertheless, the knowledge structure around the core concepts of object orientation is constantly evolving throughout the lecture. Also, the concepts of inheritance are appearing over time - they are introduced rather late in the lecture, the concept of *subclass* for example wasn't present in the material until the last measurement, but appears earlier, already. Looking at Fig. 30 it can be seen, that the development of the structural knowledge is only one part of the picture though, as the misconceptions also show a considerable development.

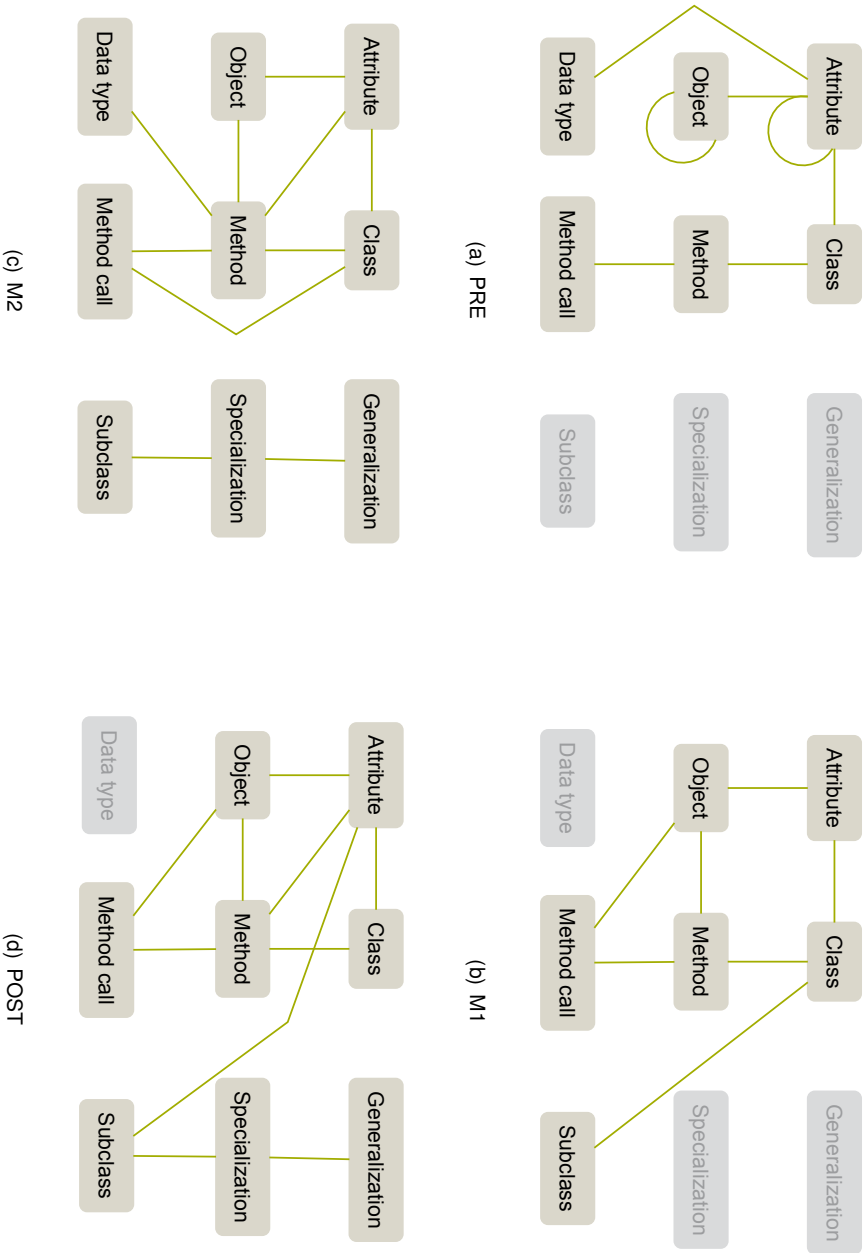


Fig. 29: Development of the high scoring propositions concerning OO related concepts.



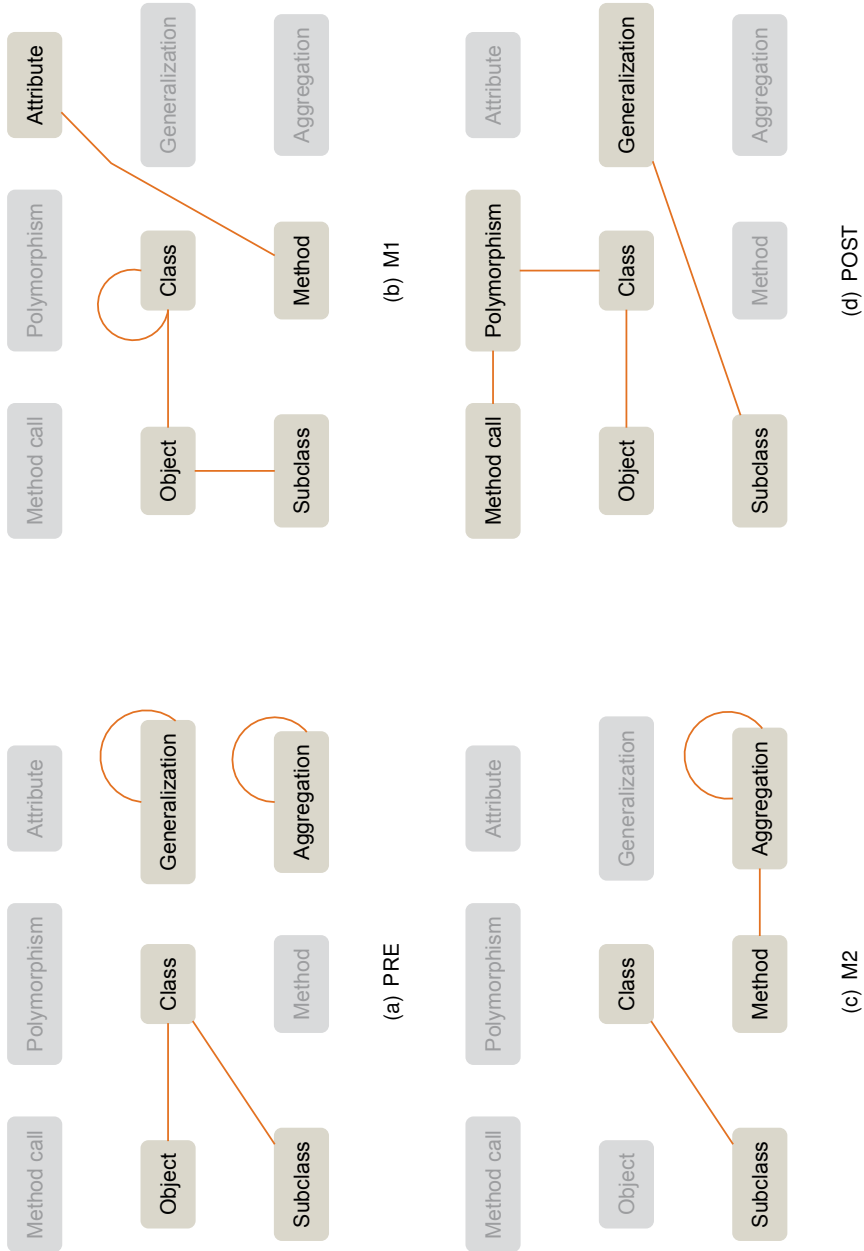


Fig. 30: Development of the low scoring propositions (misconceptions) concerning OO related concepts.

### 10.3.3 RQ3: The Process of Learning

To investigate how the learning process might reflect in the concept maps, the previous results of the development of knowledge and misconceptions will be used in the discussion. Additionally, the correlation between map scores and scores in the exercise were calculated. The map scores were formed by summing the scores of all propositions of a map. The correlation for the four points of measurement (PRE, M1, M2, POST) and using Spearman's rank correlation are: 0.28, 0.42, 0.21, 0.18. The value is distinctly different for M1.

## 10.4 Discussion

Concerning the first research question (RQ1), there are several points worth mentioning. First, as has been alluded to before, the maps are rather sparse. In other words, adding a concept to a map is equivalent of connecting it with exactly one edge to the existing map. There are exceptions, of course, where the maps are denser graphs. Also, there seems to be a "natural" upper bound on the complexity of a concept map that students are willing to produce in the given setting (about 30 minutes time and without further personal advantages when participating), as indicated by the stagnation between the last two tests. The increase in variation over time can be an indication of an increasing spread between students with a more developed knowledge and those with only poor conceptual knowledge. Clearly, motivational aspects will also influence this result. Some students will lose interest in the concept mapping task and produce only small maps, thus the overall variance will increase. The development of the edge scores in Fig. 26 also shows, that misconceptions (meaning edges scored with 0) are not diminishing over time, in contrast to what one would hope for in a lecture. Taken together with the increasing ruggedness this indicates that the students are not seeing the connection between different parts of the topics but are creating new "islands" of connected concepts in their knowledge structure. This might be due to an approach more oriented towards rote-learning or it might provide support for the "knowledge-as-elements" theory of conceptual change as described in the second part.

The increasing ruggedness also leads to the visible linear correlation between concepts and the number of clusters. While the prototypical "good" concept map would consist of only very few clusters, many concepts and even more propositions, this result has not been attained in this study. Instead, the correlation of concepts and clusters shows that on average more concepts also incur more clusters in

the map. Also, the score does correlate well with either measure, particularly with the number of concepts. Therefore, a bigger map does on average, also contain more “right” propositions. Using the transformed scoring scheme ensures that only explicitly right or wrong propositions will influence the summed score value of a map. If it is assumed that the score indeed does reflect the actual knowledge contained in a map, then the correlations of Table 10.1 show that both the measure of concepts and of propositions seem to be an indicator that can be used to predict the overall quality of a concept map.

The subgraph frequencies are revealing that the complex fact that was intended to be learned by the students is taking on different forms in their mental models - at least when externalized with concept maps. It is interesting, that the original subgraph is not occurring anywhere in the maps. Instead, three different combinations of only three of the four concepts are occurring and one combination that connects all four (and the empty subgraph). It is interesting to note that the subgraph frequencies are also undergoing a development over the different measurements. Concerning the subgraphs of *object*, *class*, and *attribute*, the majority is missing the connection between *class* and *attribute* - which might indicate that the objects first approach of the lecture is showing indirectly - the values of attributes are describing objects.

Concerning the second research question (RQ2) as shown in Fig. 29 and Fig. 30, it is very interesting to see how the knowledge of the students seems to develop. First, there is some relevant knowledge present before the start of the course. So, some of the students know what the main concepts of the lecture were before the corresponding tests and they seem to have had a mental representation of their interconnections. However, as the material of object orientation is rather complex and completely new to many students, it is not surprising that, at the beginning (i.e. M1), when they were first presented the material, there are roughly as many “problem areas” as correct associations. However, as this trend continues throughout M2 and even into POST, it indicates problem areas of the covered material. For instance, the association between the concepts *class* and *object* remains a misconception throughout almost every test and it is never present in Fig. 29. This lends itself to the interpretation that a considerable number of students seemed to neither recall factual information about the concepts and their interrelation, nor did they seem to have gained a deeper understanding of those two very central concepts of object orientation. Another interesting example is the association between the concepts *class* and *subclass*. Basically, the understanding of this association requires also knowledge about the concept *inheritance*, which wasn’t present in the lecture until the very last test. Nevertheless, the concept of subclasses are present among the frequently right associations, even in the context of inheritance, in each test

except for PRE. It is also present in each of the corresponding graphs of the wrong associations, however, indicating that the mental representation of the concept is often faulty. For example, one of the incorrect edges between *class* and *subclass* in M2 was labeled “contains”. After the corresponding material was finally presented in the lecture (between M2 and POST) a large enough number of students still had misconceptions about the concept for it to remain in the corresponding graph of POST. Note, how the association that is frequently right is between *subclass* and *specialization* while the association that is frequently wrong is between *subclass* and *generalization*. Interestingly, the “obvious” relation between *class* and *subclass* appears only once among the frequently correct associations and twice among the frequently wrong associations. For POST, it is in neither group, though.

Concerning the third research question (RQ3), first, there is the observed correlation between map scores and exercise scores. The result is highly interesting insofar as the concepts of M1, which shows the best correlation, are most closely related to the contents of the exercise (which was held closely after M1). So, for the contents covered both in the exercise and the concept mapping task, there is an indication that the abilities of the students as a result of their learning corresponds to their scores in the concepts maps. An approach oriented towards rote-learning will most probably show higher scores in the concept maps than in the exercises. This is especially true for CS when taking into account, for example, the reasoning behind the taxonomy of Fuller et al. (2007) as presented in the second part - that there is a supposed independence between the more cognitive oriented and the more application oriented learning objectives.

The overall picture when taking both Fig. 29 and Fig. 30 into account clearly shows the process of learning. The knowledge develops increasingly while misconceptions are present but also developing for the most part. In other words: As the knowledge increases, the misconceptions shift. Using the learning model of Hay et al. (2008) as shown in Fig. 4, the overall learning pattern seems to follow the deep learning approach in which the students need a reconciliation of their knowledge structures because the prior knowledge is not robust enough. This is encouraging, as it indicates that the students were choosing to learn meaningfully even in the difficult context of non-major CS education. In total, the results show that there is a visible development in structural knowledge during the course of the lecture. However, the development is not as straight forward as a simple model of learning might suggest: The misconceptions that are present at the beginning are not simply decreasing as they are replaced with “correct” knowledge, nor is everything that is newly learned “correct”. Instead, the misconceptions are developing as part of the structural knowledge.

In conclusion, the results of this case study have shown that the learning of new CS related concepts can be made visible. While the development of misconceptions does reveal problem areas - and is therefore a valuable result in its own right - it also indicates that the students are trying to incorporate the concepts meaningfully into their mental models.

Concerning the analysis methods used, the simple graph measures are not able to grasp complex information in the data. They can serve as a quick overview, though, and at least the development of the measures over time is basically like one would expect them to be. The analysis of subgraph frequencies is well suited to take a closer look at a certain small set of concepts and the different structural combinations that are present in the maps of a landscape. Combining vertical landscapes with horizontal landscapes as a technique provides useful insights into the development of structural knowledge for the group of students as a whole - especially when visualized as a series of graphs.



# 11 | CS2: Knowledge Structures of Beginning CS Students

The theories of conceptual change and meaningful learning as presented in chapter 3 as well as models of instruction planning, like the “Berliner Modell” (cf. Riedl 2010, p. 103ff.) all stipulate that “good” learning and teaching must necessarily build upon and take into account the prior knowledge of the learner. “Prior knowledge [...] provides a framework through which new information may be organized and assimilated. This reduces the amount of information chunks to be recalled and provides association cues for accessing information from the long-term memory” (Gurlitt & Renkl 2010, p. 418). This becomes a difficult task in higher education, as typically the development of persons’ knowledge when entering institutions will be much more diverse than, for example, children entering secondary education. Especially for computer science education, this becomes particularly relevant. As has been noted in the first chapter, CS is not yet an established subject of school education in contrast to e.g. mathematics or physics. Therefore the prior knowledge is expected to be highly diverse, even more so if a part of the students actually *did* receive formal CS education in school. So, the typical prior knowledge is important information for lecturers and curricular designers. It can also be used in the development of introductory courses that address possible misconceptions or underdeveloped parts of the knowledge structures in CS - such a course is investigated in the next case study.

## 11.1 Description of the Setting

The state of Bavaria, which is the second largest state of Germany, decided in the year 2000 to introduce a new compulsory subject “Informatics” at its 405 *Gymnasiums*, starting in autumn of 2004 (cf. Müller & Hubwieser 2000). This introduction came parallel with a major restructuring of the school type, leading to a reduction of 1 year in length (from 9 years to 8 years). The two systems are called “G9” and “G8” respectively. Therefore, in the summer and winter terms of 2011, the last G9 students and the first G8 students have been entering university at the same time, making it a very unique occasion for investigating the effects of the subject on their prior knowledge.

The research of this case study was done as part of a project called AVIUS that dealt with the influence that the compulsory school subject Informatics has on the

knowledge structures of beginning CS students. Additionally, the project investigated the impression that beginning students had of computer science as a subject taking into account their personal biographical history concerning CS, following the research of Knobelsdorf (2011). The participants of AVIUS were beginning CS students who enrolled at the TU München in the summer and winter term of 2011.

The next section will briefly describe the design of the school subject since its impact on structural knowledge is the main focus of the case study.

### 11.1.1 Description of the Subject “Informatics”

The text of this section is partly taken from (Mühling et al. 2010, p. 60ff.), see also (Hubwieser 2012). The Gymnasium is a specific type of secondary school in Germany. In Bavaria, it is starting at grade 5 with children at the age of 11 and leading to a degree that allows enrolling in universities after 8 years (9 years before the restructuring) of studies. It offers four different directions of study: natural science/technology, foreign languages, economy, and music/arts. At the time of the survey it was attended by about 370.000 children. The first class of G8 and of the new subject entered grade 5 in autumn 2003 and completed the compulsory stage of both school and Informatics after grade 10 in the summer of 2009. The new subject is comprised of three stages:

- In grade 6 and 7 all students of the Gymnasium have to attend 1 compulsory lesson per week.
- In grade 9 and 10 there are 2 lessons per week, compulsory for all students that have chosen natural science/technology as their direction of study (typically about 50% of the students who have to choose direction).
- In grade 11 and 12 the students that have attended Informatics in grade 9/10 can choose an eligible course of Informatics with 3 lessons per week.

The German school system is heavily federalized. Organization, types of schools, subjects and curricula vary from state to state. However, all states have a rather strict system of curricula. A typical curriculum will explicitly state the learning objectives (including suggested time frames), that students have to learn in each grade. Thus, teachers have almost no freedom in deciding what they're teaching in their classes.

The curriculum follows to a great extent the information-oriented teaching approach as presented by Breier & Hubwieser (2002). Basically the students should acquire three different basic competencies within this new subject:



- Represent, structure, store, link, and exchange information using suitable hard- and software combinations.
- Master, describe, and communicate about complex systems using suitable - particularly object-oriented - modeling techniques.
- Implement models using suitable software systems or programming platforms - particularly object-oriented - programming languages.

Concerning teaching methods, the subject Informatics is encouraging, more than any other subject, a very modern, student-oriented learning style: the intensive problem-oriented usage of computers forces the use of teamwork, group-teaching, project work and product-orientation in the lessons. Hubwieser (2007) points to an underlying didactic dilemma: Modern teaching approaches postulate to pose authentic problems to the students that have relevance in the “real world” outside the school. Thus, it seems advisable to start programming with interesting, sufficient complex tasks that convince the students that the concepts they have to learn are really helpful for their later professional lives. Hence we have to start our programming course with quite complex programs that simulate processes that the students know from their everyday life. On the other hand, if we start with quite complex object-oriented programs, we might ask too much of the students, because they will have to learn an enormous amount of new, partly very difficult concepts at one time. As Hubwieser (2008) states, a solution to this problem is to follow a very strict “objects first” teaching strategy (cf. Gries 2008).

The basic idea is to start the course in grade 6 (where the students are 11 or 12 years old) with object modeling of standard software documents like vector and pixel graphics, texts or multimedia presentations or hypertext structures. This way, the students learn to use the object-oriented concepts *object*, *attribute*, *class*, *method*, *association*, *aggregation* and *reference* in order to manipulate documents, some years before they will have to apply them in the context of object-oriented programming. In grade 6 the students start working with objects of classes like *circle*, *rectangle*, *symbol*, *paragraph* etc. They find out that some of the objects are connected by aggregations, which can even be recursive, for example *folder* within file systems. In grade 7 they learn to apply the concept of references (implemented as links) in order to construct hypertext structures and to exchange information using e-mail systems. At the end of grade 7, they learn to activate objects by programming their own methods, using simple robot systems.

At the beginning of grade 9 the students apply the concept of *functions* by designing data flow diagrams, which are then implemented using spreadsheets. Following this, they construct object-oriented data models and implement them using relational

databases. In grade 10, they finally start “real” object-oriented programming and thereby have to apply all the concepts they have learned in the former grades.

In the eligible course of grades 11 and 12, the students work with recursive data structures (lists, trees, and graphs) and corresponding recursive algorithms. They apply the basic concepts of software engineering: software life cycle models and stages of software development. In grade 12, they design formal languages and learn how parallel processes can be synchronized. They simulate computer networks, analyze their topology, and learn how a computer is working principally (hardware architecture, register machine, etc.). Finally, they have to accept that there are limits of computability, with all the consequences for e.g. data security or cryptology.

## 11.2 Data Collection & Research Questions

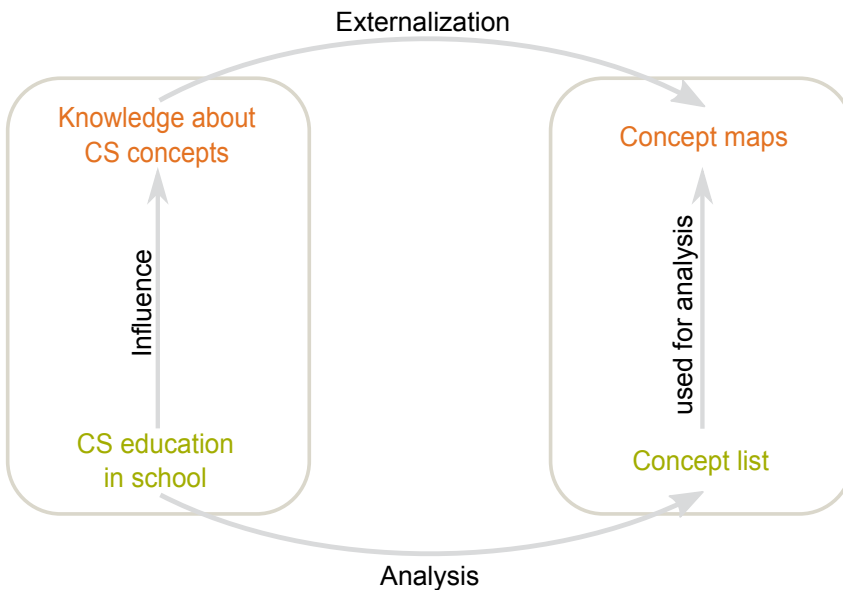


Fig. 31: Overview over the study, based on the schema of Fig. 1.

The study is a cross-sectional study of the beginning students of the TU München. The relevant collected data consists of personal data of the students - including the number of years they received CS education in school - and a concept map they were asked to draw. Additionally, they were asked to provide free text answers concerning their CS related biography, but the answers are not used in this case study.

The data collection itself happened at several occasions: All of the newly enrolled students were invited to take part in a preparatory course in mathematics and object-oriented programming (see next chapter). Also, their first two days at university are organized as an introductory event before the start of the lectures at the third day of the term. The participation in this study was integrated in all of these events as part of the regular schedule of the students, i.e. they weren't asked to voluntarily participate in their spare time, but instead the whole group was given the survey and asked to complete it, even though there were no negative effects for students who chose to not take part.

There were 590 participants of about 700 new students who were assessed in 4 batches according to their schedule of their first days at the university. After eliminating all useless survey responses, 338 remained.

The participants were given a list of 40 concepts (see below). They were then first asked to mark all concepts they were familiar with and afterwards draw a concept map of those. Several different orderings for the concept list were used in order to measure whether this has an effect on the maps, but none was found. The focus question was to begin with connecting a pair of concepts that has been marked as familiar and then incorporating as many of the marked concepts as possible by named links. The students received a short written introduction on concept mapping including an example map. The participants were asked to label the propositions. The maps were produced with pen and paper and then later digitalized for analysis. For the whole survey, the students were given 45 minutes time. An example of one of the maps can be seen in Fig. 32. In each room there was a researcher present (at least for some time at the beginning) who told the students about the goals of the survey and about the anonymity of the process. They also addressed questions of the participants.

The concepts were extracted from the curriculum of the school subject semi-automatically by extracting nouns, counting their frequency and then manually filtering the list. The 40 concepts are:

*algorithm, array, assignment, association, attribute, automaton, class, conditional statement, data, data structure, edges, flow, grammar, graph, instruction cycle, list, loop, method, object, processor, program, programming language, record, recursion, register machine, semantics, sequence, state, state change, statement, subclass, superclass, syntax, table, tree, pointer, variable, value, vertex, working memory*

When using ConEx - like in the last chapter - and extracting the first 40 concepts except for different word forms, there is an agreement of just 50%. However, the

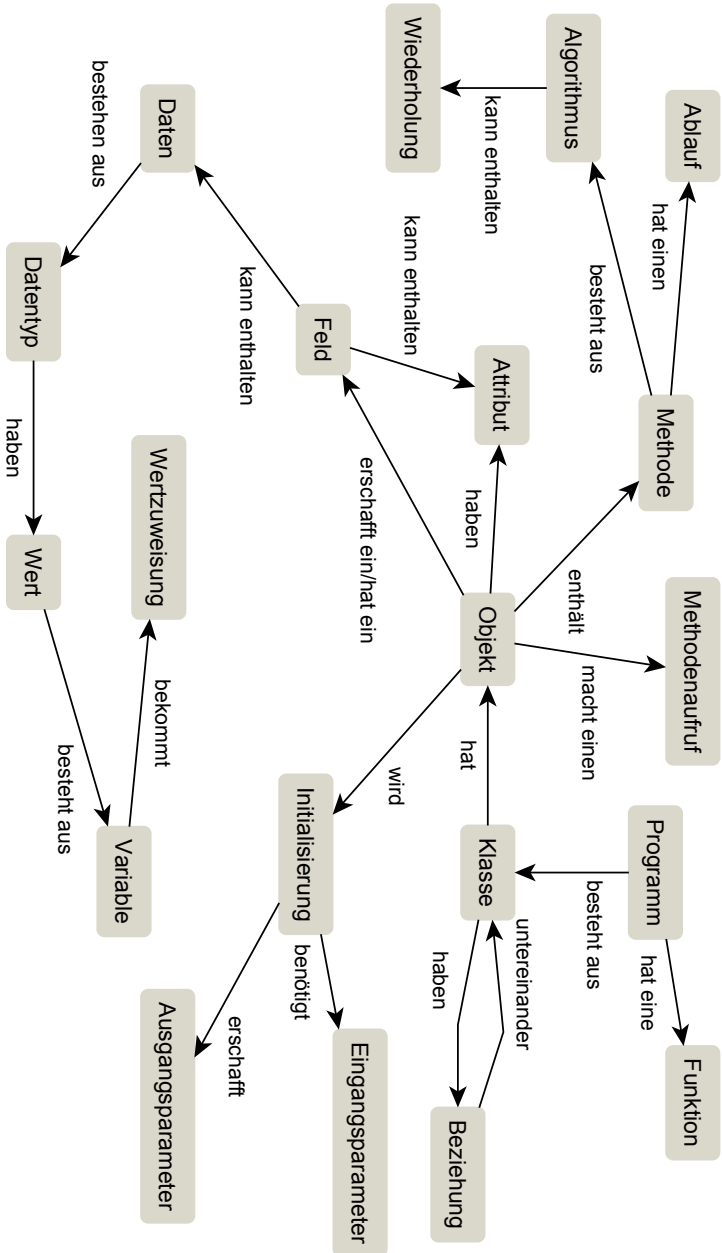


Fig. 32: A digitalized example map originally produced by one of the students in German.

curriculum is not a very good input for a *completely* automated extraction as there are many special words like “pupil” and “hour” that occur rather frequently and therefore interfere with the result.

Fig. 31 shows how this particular study related to the schema of Fig. 1. The research questions are:

**RQ 1** What is the prior knowledge of a beginning CS student in general and are there general misconceptions?

**RQ 2** Is it possible to observe an effect of the newly introduced subject Informatics in Bavarian secondary schools with regard to the knowledge structures of beginning CS students?

Concerning the second question, there are other ways in which the subject may have had an impact of course, for example by bringing students to enroll for computer science who otherwise wouldn't have chosen this field of study. The research project AVIUS is concerned with the broader scope of this impact, but this case study is only concerned with the aspects relevant for this thesis.

## 11.3 Analysis and Results

### 11.3.1 RQ1: Prior Knowledge of Beginning CS Students

To arrive at a visual representation of the “common” prior knowledge, all of the concept maps were amalgamated vertically. Edge weights of the amalgamated graph were calculated by counting the original edges in order to use Pathfinder analysis. The maps were restricted to the 40 concepts of the list. Additionally, edges not occurring in at least 10% of the maps were removed manually before creating the Pathfinder network. To arrive at the sparsest representation, the parameter values  $q = 39$  and  $r = \infty$  were used. The Pathfinder network is shown in Fig. 33, the concepts *assignment*, *association*, *subclass*, and *superclass* were left unconnected and removed. The five most central concepts according to the measure of betweenness-centrality in descending order are: *program*, *processor*, *working memory*, *data* and *class*. Connectivity measures for sparse Pathfinder networks are not very informative, as they will naturally be low. The four nodes with the highest appearing node degree (4) are *program* and *processor* again as well as *data structure* and *programming language*.

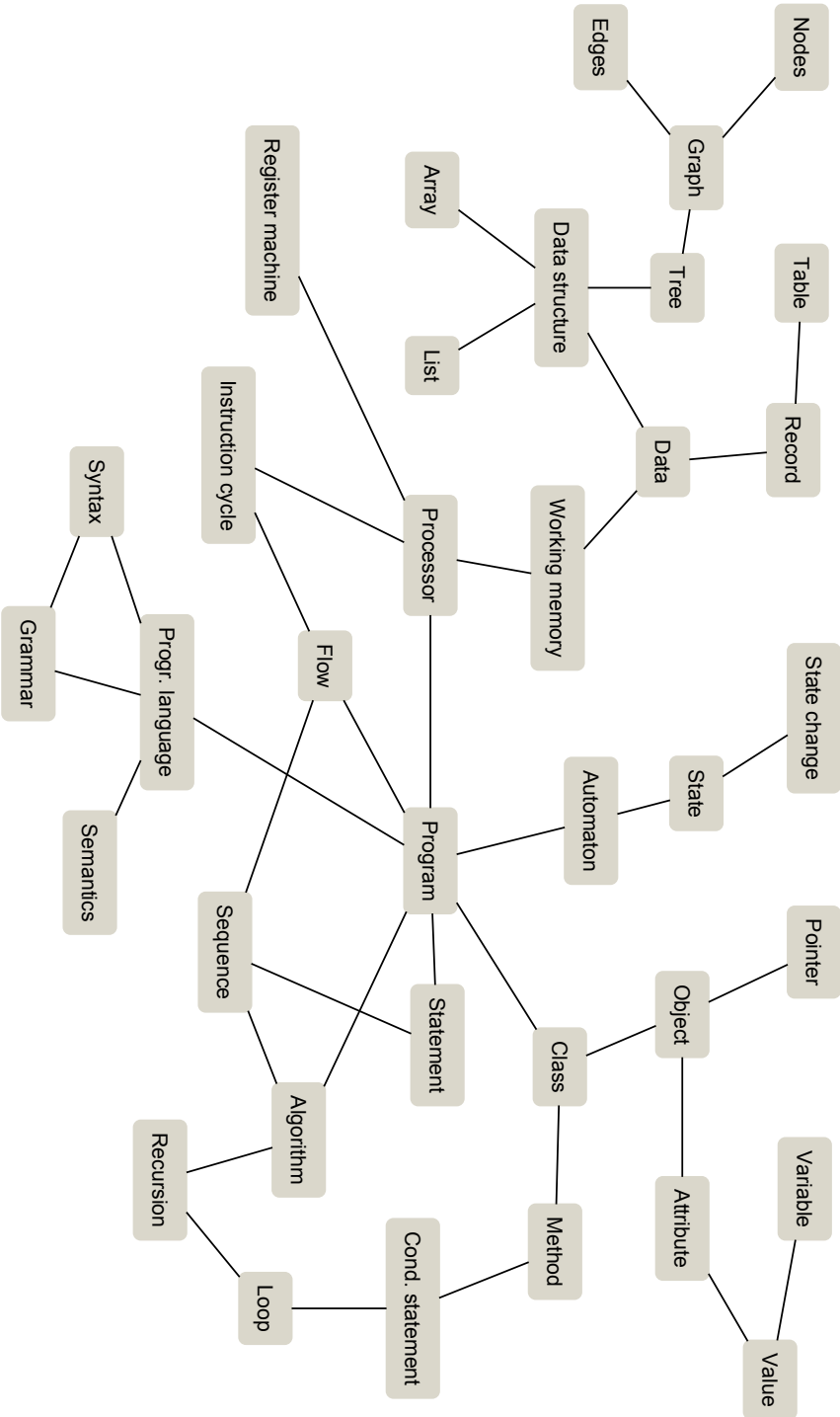


Fig. 33: The Pathfinder network of all the maps.

Next, a cluster analysis was used in order to find out whether or not there are structural differences inherent in the data. The concept maps were accumulated vertically in two different ways. First, the concept matrix was used together with the MBMM approach, then, the graph similarity matrix and the PAM algorithm were used (see section 7.2). In the latter case, the Hopkins index (see section 5.3.1) of the resulting matrix was 0.83 indicating that a clustering of the data is possible.

The MBMM clustering applied to the concept matrix finds an optimal solution at three clusters. One of the clusters, containing 81 maps, simply encompasses all empty or nearly empty maps. The maximal relative frequency of occurrence for all concepts - i.e. the sum of each column of the concept matrix of this cluster divided by the size of the cluster - is a very small 0.03. The mean relative frequency over all concepts is just 0.004. Also, judging by the overall probabilities, the concept maps of the remaining two clusters containing 100 and 151 maps, are predominantly differing in complexity. This is confirmed by a t-test. The true difference in means (of the edge count) between the two groups is more than 7 ( $p = 0.0007$ ). This amounts to an increase in edges of more than 70% for the maps of one cluster compared to the other.

Using the PAM algorithm applied to the distance matrix of graph similarities, the optimal number of clusters is two. Both clusters are of nearly identical size with 169 and 164 maps, respectively. To identify how the maps of both clusters differ, each set of maps was amalgamated vertically and the Pathfinder networks of both landscapes using  $q = 39$  and  $r = \infty$  were created. Again, all edges that appeared in less than 10% of the maps were removed beforehand and unconnected concepts removed. Fig. 34 and Fig. 35 shows how both clusters organize their knowledge.

The structural configuration of many concepts is alike, as the connections between the concepts *program*, *processor*, *register machine*, *working memory* and *data* show. But there are also many differences. Cluster 1, in contrast to cluster 2, has *data* not connected to *data structure*, for example. Also, the concept *recursion* is missing completely in cluster 2 and in cluster 1 it is connected to the concepts *algorithm* and *tree*. Also, cluster 1 has the concepts *conditional statement*, *loop*, and *algorithm* all connected to *method*, whereas cluster 2 has *conditional statement* and *loop* connected to *algorithm*, which in turn is connected to *program*. The concept *method* appears only as an "appendix" to *class* for this cluster. The concept *statement* is, interestingly, connected to *program* for both clusters, but only cluster 1 shows an additional connection (*flow*).

Finally, all the available data of the students was used in order to find out whether or not the persons forming the two clusters differ significantly in any of the avail-

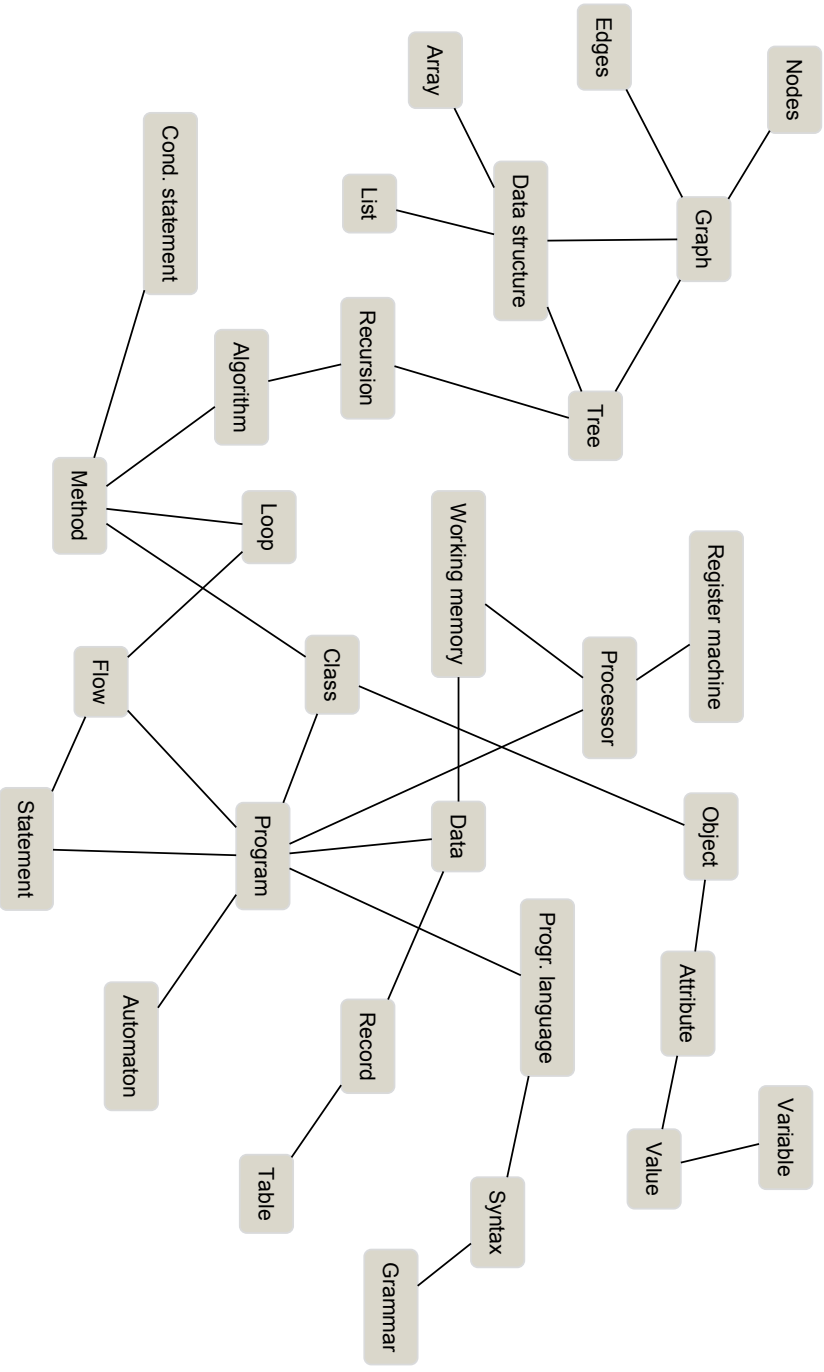


Fig. 34: The Pathfinder network of the first cluster identified by the PAM algorithm.



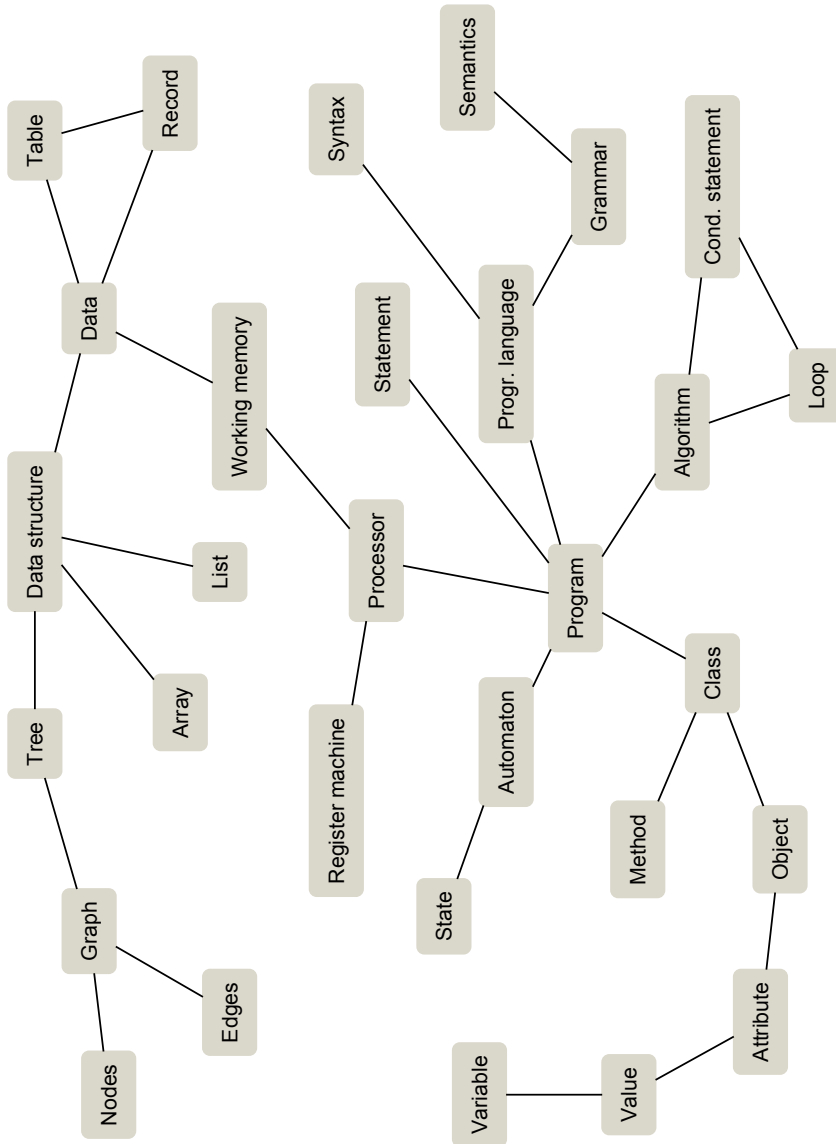


Fig. 35: The Pathfinder network of the second cluster identified by the PAM algorithm.

able attributes. There is no discernible difference concerning whether or not they attended the G8 or G9 school system, as a  $\chi^2$  test reveals. 49.7% of the students of the first cluster were attending the G8 system and 45.1% of the second cluster. Also, there are no differences concerning whether or not computer science was one of their subjects when they graduated from school or concerning the number of years they received CS education in school. However, a  $\chi^2$  test does show a significant difference ( $p < 0.05$ ) concerning who they think their biggest influence regarding their interest for CS was. The students of cluster 1 have given their father as the biggest influence significantly more often than the students of cluster 2.

### 11.3.2 RQ2: Effect of CS Education in Secondary Schools

For this analysis, the maps were split into two groups according to the students' prior education. *G8* denotes the group of students who attended the compulsory subject Informatics for at least four years and *G9* denotes the group who didn't have a compulsory subject. Some of them had a voluntary subject of computer science as part of their education though. Also, in the *G8* group, there are some who took Informatics as one of their subjects they graduated in and others who didn't. Note that the groups do not form a partition of the data as there are 48 cases of students who either attended some other form of school, came from another country, or whose answer to the survey could not be clearly attributed to either *G8* or *G9*. The *G8* group consists of 163 maps and the *G9* group of 127. To investigate the impact, it is reasonable to separately aggregate the concept maps of these "naturally" occurring clusters and check whether or not there are discernible differences. The maps of the *G8* groups are significantly denser: A t-test of the number of edges between two groups shows, the hypothesis, that the true difference in means is 0 can be rejected with a confidence level of 99% ( $p = 0.0001$ ).

To identify the prevalent structural configuration, both sets of maps were amalgamated vertically by summing edges and, again, the Pathfinder networks with parameters  $q = 39$  and  $r = \infty$  were created. As before, edges that were appearing in less than 10% of the maps of each group were removed beforehand and unconnected concepts were removed. First, taking a look at the concepts that remained in the Pathfinder networks, there were 31 for *G8* and 26 for *G9*. The networks are shown in Fig. 36 and Fig. 37 respectively. All concepts appearing in the *G9* networks are also appearing in the *G8* network, however, the *G8* network contains the additional concepts: *list*, *recursion*, *semantics*, *state*, and *register machine*. When measuring the betweenness-centrality of the nodes, the three highest

Community	G8	G9
1	<i>Data structure, Graph, Edges, Nodes, List, Tree, Array, Recursion</i>	<i>Data structure, Graph, Edges, Nodes, Tree, Array</i>
2	<i>Data, Record, Table, Working memory, Processor, Register machine</i>	<i>Data, Record, Table, Working memory, Processor</i>
3	<i>Object, Attribute, Value, Variable</i>	<i>Object, Attribute, Method, Algorithm</i>
4	<i>Class, Method, Loop, Conditional Statement</i>	<i>Class, Value, Variable, Loop, Conditional Statement</i>
5	<i>Programming language, Syntax, Semantics, Grammar</i>	<i>Programming language, Syntax, Grammar</i>
6	<i>Statement, Program, Automaton, State, Algorithm</i>	<i>Statement, Program, Automaton</i>

Table 11.1: The communities identified within the Pathfinder network from the maps of the G8 and G9 groups. The ordering of the communities for both groups is arbitrary, but was chosen to allow comparing the groups more easily.

scoring concepts for the G8 group are: *program*, *class*, and *data structure*. For the G9 group it is: *program*, *processor*, and *data*.

Analyzing the communities in the Pathfinder networks using a greedy algorithm, both networks are partitioned into six communities. The concepts are assigned to the communities as shown in Table 11.1.

## 11.4 Discussion

For the first research question (RQ1), the visual inspection of the Pathfinder network as shown in Fig. 33 reveals some interesting insights. First, it is peculiar that, on average, a beginning CS student does know something about the concept of a register machine or the theoretical constructs of grammars, but not of the concepts of subclass/superclass. It can be assumed that the knowledge about register machines is most probably due to the large amount of G8 students and therefore an artefact of prior formal CS education. This nevertheless points to a not fully developed prior knowledge regarding core OO concepts - even more since subclass and superclass

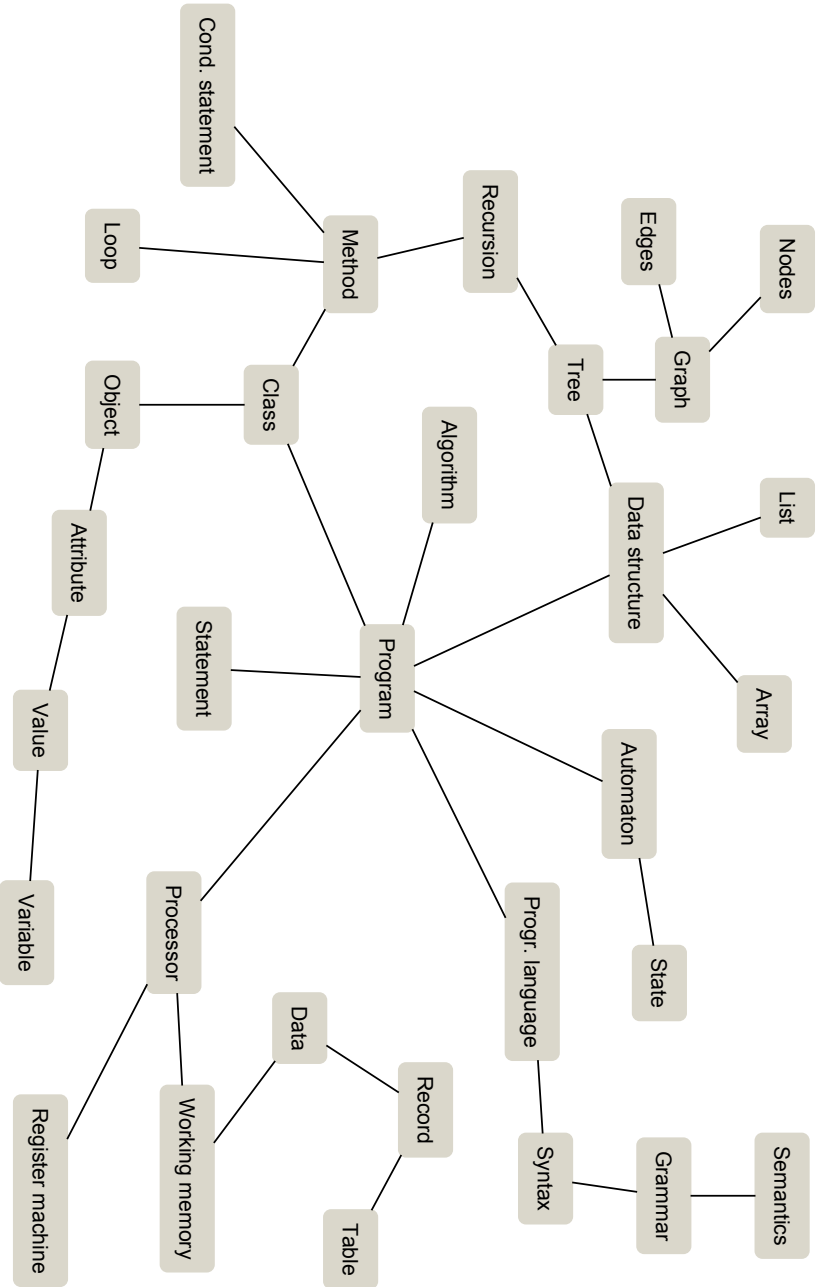


Fig. 36: The Pathfinder networks of those students who attended the compulsory school subject.

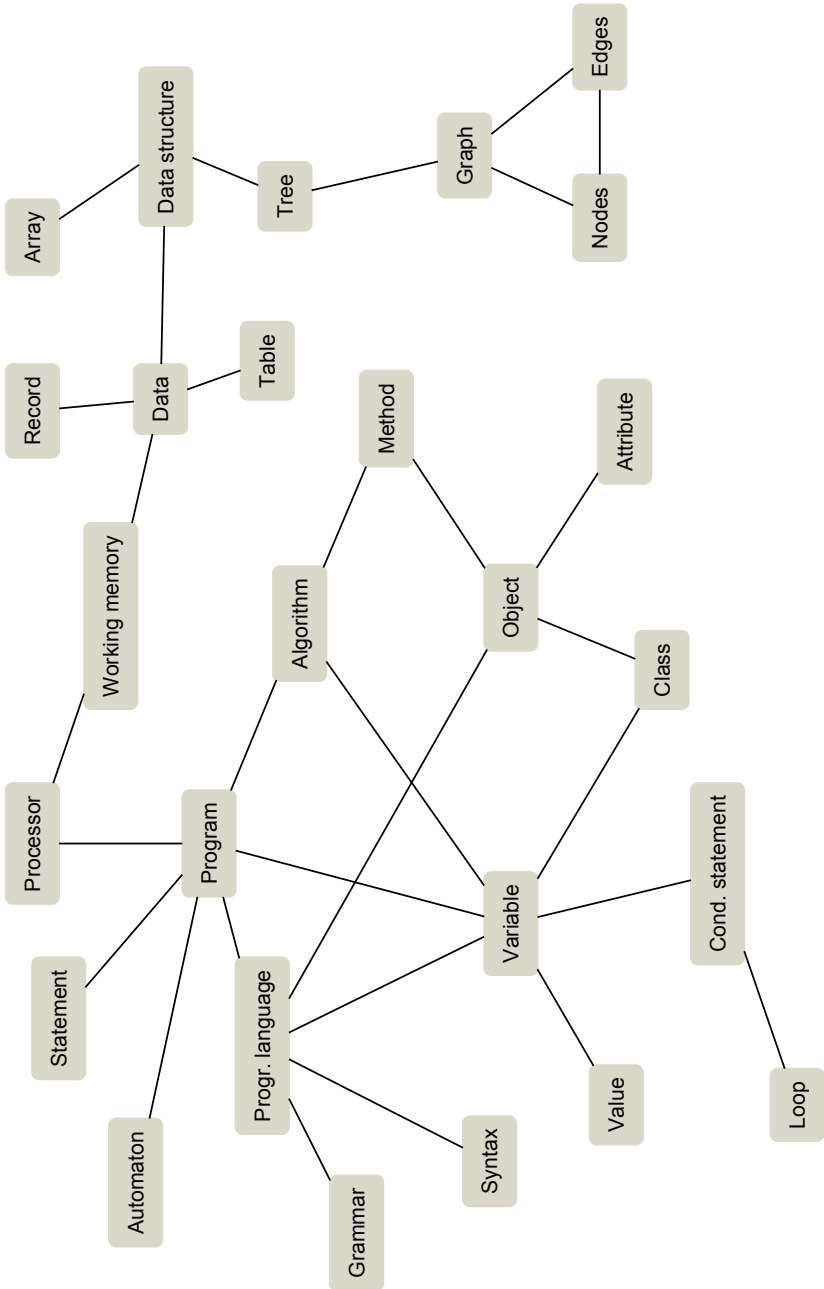


Fig. 37: The Pathfinder networks of students with no compulsory school subject.

are also central concepts of the curriculum of the school subject. Taken together with the fact that the structural configuration is centered around the concepts of *program* and *processor*, and also taking into account the associations between *data* and *working memory*, as well as between *register machine* and *processor* all indicate an understanding of CS that is centered around computers and a more practical, programming-oriented approach towards CS. An expert network would most probably differ in the connections regarding these concepts and show a more distinct separation between general and abstract concepts like *register machine* or *data* and specific, technical concepts like *processor* or *working memory*. This is valuable information for the design of introductory courses and lectures.

The cluster analysis shows, however, that there are also hidden structural differences in the groups. Based on the structural configuration of the concepts, the persons of the two clusters can be characterized as follows:

**Cluster 1** Seem to be fluent with the object-oriented approach to programming, since *algorithm* is connected to *method* as well as the control structures *loop* and *conditional statement*. Also, they have an understanding of recursion and recursive data structures. The (recursive) data structures *list*, *array*, *tree*, and *graph* are forming a group of concepts that is far away from the group of database oriented concepts (*data*, *table*, and *record*), which are placed near the technical concepts *working memory* and *processor*.

**Cluster 2** Are clearly more oriented towards procedural programming, as *algorithm* is connected to *program* and the control structures *loop* and *conditional statement* are connected to *algorithm* in turn. The concept *method* is seemingly only weakly integrated into the mental models of the persons of this cluster. Also, the concept of *recursion* is missing completely. The three database oriented concepts and the data structures are forming a large group of concepts for this cluster, which is only connected to the rest via the concept *working memory* (i.e. this is the strongest link that remained in the Pathfinder network).

So, the central aspect that differs between these clusters seems to be their approach (or prior experience) towards programming - object-oriented versus procedural - and their grasp of the concept of recursion. Since recursion is a central concept of computer science and its presence or absence has a clear structural effect on the knowledge structures, this may be taken as a sign that it is a threshold concept of computer science. It is interesting to see that the school type did not show in this particular clustering - especially when comparing some obvious structural

similarities to the Pathfinder network of the G8 group shown in Fig. 36. In how far the difference of the father as an influence affects the results is unclear. It seems reasonable to assume that in these cases the father has either studied computer science or is at least working in this area. This may explain that these students know more about object orientation and recursion as they possibly had better access to CS learning material or may have learned something directly from their father.

Concerning the second research question (RQ2) the results are clearly showing artifacts of the specific curriculum of the school subject with its strict objects-first, object-oriented approach. Taking together the Pathfinder networks of Fig. 36 and Fig. 37 as well as the communities of Table 11.1, the following can be observed:

- The G8 network is visibly more complex.
- While it seems common among all beginning students to value the concept *program* highly in their knowledge structure, the most central concepts of *class* and *data structure* as opposed to *processor* and *data* show a more object-oriented understanding of the G8 group, while the G9 group seems to be more focused on the technical aspects and computers themselves.
- The network of the G8 group has a connection between *recursion* and *tree*, which corresponds to the approach of introducing recursion based on object-oriented recursive data structures, like lists and trees, chosen in the curriculum.
- Another indicator for a more formal CS education in the G8 group: For their network, there is the path *programming language*, *syntax*, *grammar*, *semantics*, whereas the G9 group only has *grammar* and *syntax* connected to *programming language* directly (*semantics* is missing completely).
- For the G8 group, communities 3 and 4 seem to indicate that objects and classes are seen as somewhat unrelated, with objects being more identified by their attributes and classes more by their methods. Additionally, the control structures (*loop* and *conditional statement*) are grouped with *class*. This resembles, to some degree, the approach of the school subject, where objects are introduced as entities described by the values of their attributes right from the beginning and later on methods are implemented for classes.
- The G9 group in contrast, has a less clear cut grouping concerning the core concepts of object orientation. Even though *method* is grouped with *object* and *algorithm*, it is again *class* that resides in the same community as the control structures. Also, *value* is not in the same community as *object*.

- There are some commonalities between the two groups: First, communities 1, 2, and 5 are nearly identical except for concepts missing the G9 network altogether. Next, there is a clear grouping of concepts related to data structures for both groups in the first community. Also, for both groups, the more database oriented concepts (*data*, *record*, *table*) are grouped with *processor* and *working memory*, placing them in a more technical and less abstract corner. In the same vein, *register machine* is seemingly more related to a real processor than to an abstract notion for beginning students. Finally, for both groups, *automaton*, *program* and *state* are related; however, none of the object-oriented concepts are put into that group, indicating a lack of understanding of the semantics of object orientation.

In conclusion, the beginning students of this case study have a surprisingly complex prior knowledge of computer science that seems - without a formal education in secondary school - to exhibit a practical, programming-oriented view on computer science. The school subject has a clearly visible impact on the conceptual knowledge, however: Investigating the Pathfinder networks of both groups, there are several indicators that point to the success of the object-oriented, strictly objects-first approach of the school subject in the sense that it has an identifiable impact on the knowledge structure of the students. The communities also support these findings.

The analysis methods used in this study were mostly centered around Pathfinder networks and their subsequent analysis using graph measures and manual, visual inspection. The methods were well suited to find out the differences in the knowledge structures between the two groups of G8 and G9 students as well as to extract the common knowledge of a beginning CS student. Also, the clustering using graph similarities revealed some interesting insights beyond the separation of G8 and G9.



# 12 | CS3: Conceptual Knowledge and Abilities

Most of the results of this chapter have been previously published in (Berges et al. 2012).

Students beginning their studies of computer science often have difficulties in learning programming, especially object-oriented programming (OOP) in the fast pace that the curriculum at university demands. Teaching and learning OOP is inherently difficult, as the research literature, e.g. (Hubwieser 2008), (Eckerdal 2006), (Ragonis & Ben-Ari 2005), shows.

Since being able to program is typically not required for students who enroll for studies of computer science, there are many with no prior experience in programming, making the beginning of their studies even more difficult. To improve their chances and alleviate the differences between beginning students, the Department of computer science at the TU München has decided in 2008 to design and implement a two and a half day long, voluntary introductory course that is held right before the beginning of the semester. This course introduces the basics of object-oriented programming to all students wishing to participate. Every newly enrolled CS student receives an invitation. As part of this course, the knowledge of the students regarding concepts of object-oriented programming has been assessed by concept maps in a pretest and posttest, following the observation of Novak & Cañas (2010, p. 3): “When concept maps are used as pretests and then employed with new learning material, research has shown that meaningful learning can be much facilitated”.

## 12.1 Description of the Setting

The design of the course is described in detail by Hubwieser & Berges (2011). It is based on the premise of Constructivism and minimal instruction. The students are given a programming task to solve and are encouraged to try programming on their own, right from the beginning with only minimal input given to them in the form of four worksheets: The first sheet describes the task itself and the software that will be used, so the students receive a short overview over the course. The second sheet introduces the basic concepts of object orientation: objects, classes, attributes, and methods. It emphasizes the concept of data encapsulation/information hiding that the students should adhere to as early as

possible. The third sheet presents the implementation of those concepts in Java. Java was chosen as the programming language, as this is the language they will be mainly dealing with in the first semesters. The last sheet presents the concept of algorithms as well as the control structures of procedural programming (sequences, conditional statements, loops). In addition to the worksheets, a peer tutor is also present to guide them whenever necessary.

The setting was developed to encourage attaining basic programming abilities without requiring much conceptual knowledge to be learned beforehand (as the conceptual basis will be presented in the lectures starting right after the course anyway). Nevertheless, it seems reasonable to assume that the conceptual framework of the students concerning the concepts of programming will change nevertheless, throughout this course.

The students worked in small groups based on their prior programming experience. There were three different levels and the students were assigned based on their responses to a question in the survey (see below) concerning their experience:

1. "I have no experience at all".
2. "I have already written programs".
3. "I have already written object-oriented programs".

The demands of the programs that the students should realize differed according to their respective level of programming experience. The students of the first level (with the least amount of prior experience) were asked to program the game "Mastermind". The groups of the next level should create a tool for managing results from a sports tournament, for example a football league. The groups of the most advanced, third level were given the task to program a version of the dice game "Yahzee".

## 12.2 Data Collection & Research Questions

All data was collected right before the winter term of 2010/2011. It consists of a concept map drawn directly before the start of the course (*pre map*) and one drawn right after the course (*post map*). Additionally, the results of a survey asking for some personal information and prior programming experiences were available for each participant. Finally, the source code that each student produced was collected and used for analysis. Drawing the concept map and answering the survey was part of the course, however participation in the course was voluntary and dropping

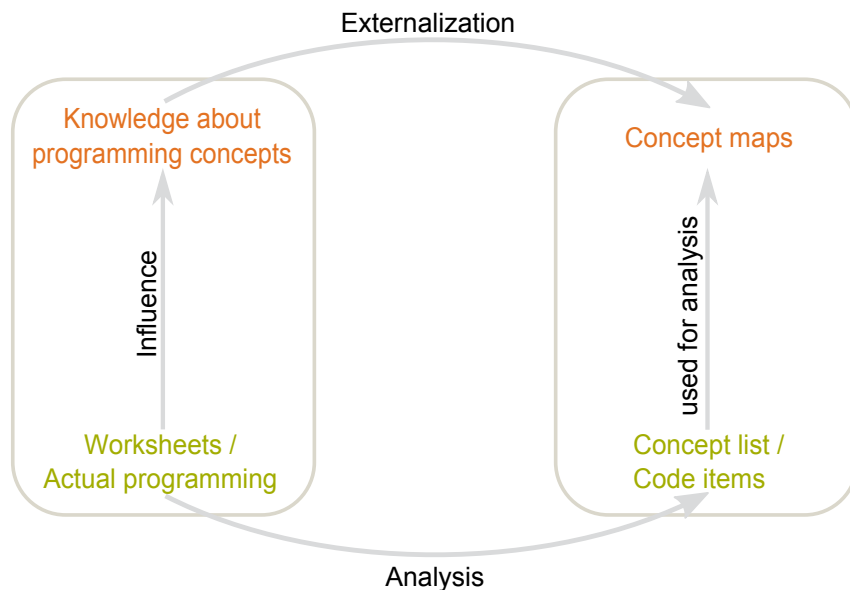


Fig. 38: Overview over the study, based on the schema of Fig. 1.

out of the course didn't incur any negative consequences for the students. Also, all data was collected anonymously. A number was given to the participants in order to match the single items of each student.

There were 167 participants, which amounts to about 42% of all newly enrolled CS students at the department of computer science at the TU München. They were divided into 18 groups ranging in size from 6 to 15 people. Some students dropped out before the end of the course and some didn't provide a complete data set of survey, source code and two concept maps. In cases where a post map was present, but not a pre map, it was assumed that the pre map was intentionally left blank due to no relevant prior knowledge. In the end, the data of 75 students could be used for analysis.

The concept mapping task consisted of a list of concepts given to the students and a written introduction on concept mapping. Additionally, a tutor was present whom the students could ask. The students were given 30 minutes for drawing the concept maps both times. Drawing was done using pen and paper. The focus question asked the students to try to create a concept map using the concepts from the given list that they were familiar with. It explicitly asked for finding labeled connections between pairs of concepts and stated that not each pair of concepts must be related. An example of one of the maps can be seen in Fig. 39. The list of 21 concepts (CL) that was given to the participants of the course was extracted systematically from

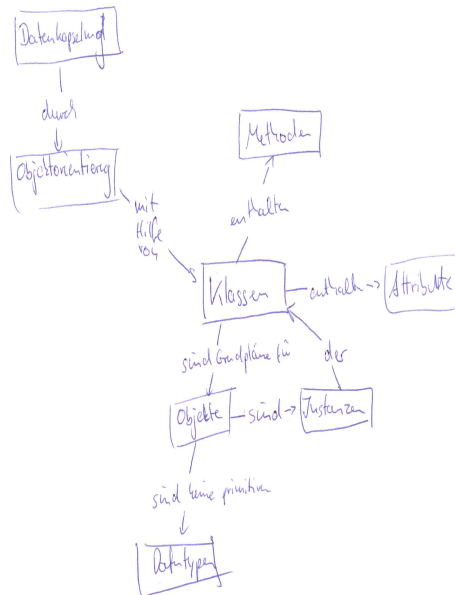


Fig. 39: A scanned example map after that was produced by one of the students in German.

the worksheets and encompasses all concepts that the course is dealing with. It is shown in Table 12.1, the abbreviations are used in the next section.

The source code was collected by asking the students to upload their code as an archive to a server. The tutors assisted in the process. The students started by using *BlueJ*<sup>1</sup> for the reasons given by Bergin, Bruce & Kölling (2005), but were then encouraged to switch to a more advanced IDE (*Eclipse*<sup>2</sup> or *Netbeans*<sup>3</sup>).

Fig. 38 shows how the schematic organization of this particular study looks like. The two research questions concerning this thesis are:

**RQ 1** How will the conceptual knowledge of concepts related to programming develop, if there is only very little actual input given. In other words, without enough material and time to rote learn but opportunities to apply the concepts in programming, will there still be a visible development in knowledge?

**RQ 2** Is there any identifiable relation between the conceptual knowledge and the actual programming abilities of the students?

<sup>1</sup><http://www.bluej.org>

<sup>2</sup><http://www.eclipse.org>

<sup>3</sup><http://netbeans.org>

<b>AM</b>	access modifier		constructor	<b>ME</b>	method
<b>AR</b>	arrays	<b>DE</b>	data encapsulation	<b>OB</b>	object
<b>AG</b>	assignment		data type		object orientation
<b>AC</b>	association	<b>IN</b>	inheritance	<b>OP</b>	operators
<b>AT</b>	attribute	<b>IZ</b>	initialization	<b>OV</b>	overloading
	class		instance	<b>PA</b>	parameter
<b>CS</b>	conditional state- ment	<b>LO</b>	loop statement	<b>ST</b>	state

Table 12.1: The 21 concepts that the students should use for drawing the concept maps. The abbreviations are used later for the code analysis and in the diagrams of the results.

## 12.3 Analysis and Results

For analysis, the data was divided according to whether or not a person had prior programming experience. There was no differentiation made between those who had experience with object orientation and those who only had written non-OO programs to keep the two groups roughly the same size: From the 75 participants that remained for analysis, 42 students had prior programming experience while 33 students had never programmed before.

The concepts of the maps were restricted to the 21 from CL given to the participants. Missing concepts from CL were added as isolated nodes to the concept map before aggregating and additional concepts not on CL were removed. Also, edges with no label were excluded from analysis altogether. The remaining propositions were then scored manually according to the following scheme:

- If the proposition forms a correct statement it will be scored with 2.
- If the proposition forms a statement that is clearly wrong or if the meaning of the statement cannot be understood, it will be scored with 0.
- If none of these two conditions apply, the association will be scored with 1.

To validate the reliability of the grading scheme three experts were asked to grade a randomly chosen subset of the edges according to the scheme. The correlation of grades between any pair of the three persons was constantly at or above 0.8. To further improve the reliability of the grading for analysis, propositions with a score of 1 (i.e. the “unclear” cases) were excluded for subsequent analysis steps.

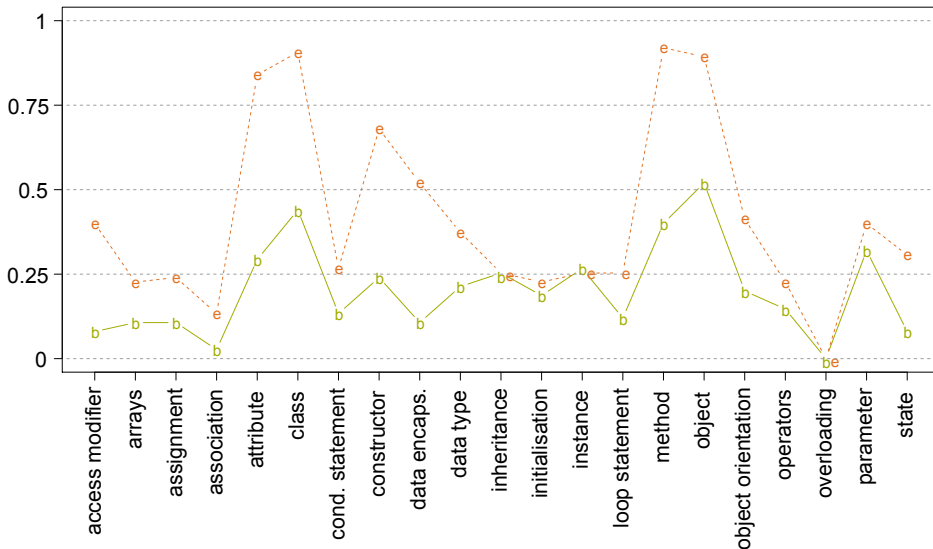


Fig. 40: Development of relevant structural knowledge over all participants. Shown is the fraction of maps with at least one “correct” edge (i.e. with a score of 2) incident to a concept (**b**(eginning) = pre test, **e**(nd) = post test).

### 12.3.1 RQ1: Development of Structural Knowledge

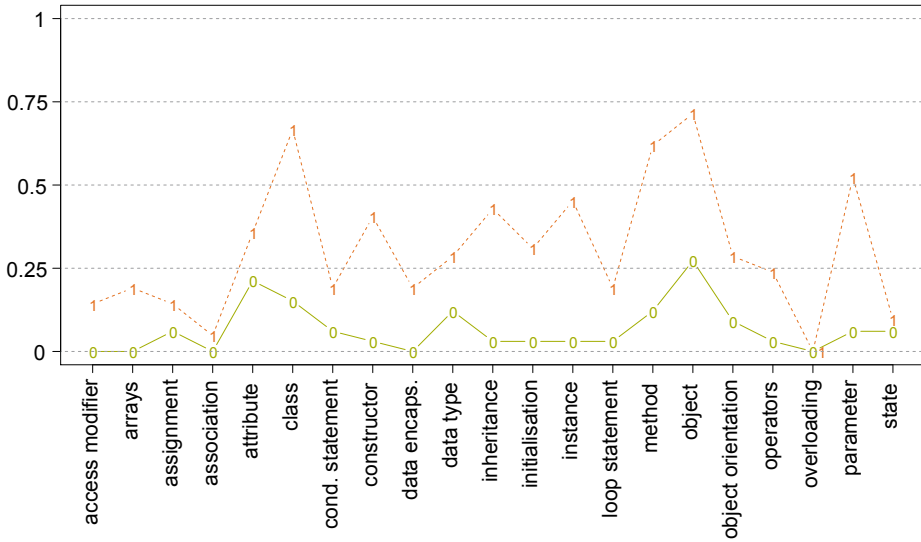
First, the concept maps were accumulated vertically - for pre and post test - both for all students combined and separately for students with and without prior programming experience. The accumulation used the concept matrix with the restriction that a concept is marked as present (value 1) in each concept vector only if at least one of the incident edges of this concept has a score value of 2, i.e. is a “correct” proposition representing some accepted “fact” about a particular concept. Then, the mean of each column was taken, resulting in a vector that contains for each concept of CL the relative frequency of this concept to appear in a map of the landscape with a correct incident edge. Clearly, this is a very basic measure and one might expect that the percentage of maps fulfilling this criterion for a given concept is very large. However, as the concept maps were rather small and sparse, it turned out to work well in practice.

The relative frequencies and especially the development from pre to post test can then be plotted. Fig. 40 shows the landscape of all maps, regardless of prior experience. Additionally, Fig. 50 in the appendix shows the “delta” (i.e. the difference) between the two lines for better readability. There is a clearly visible development of the structural knowledge over the course.

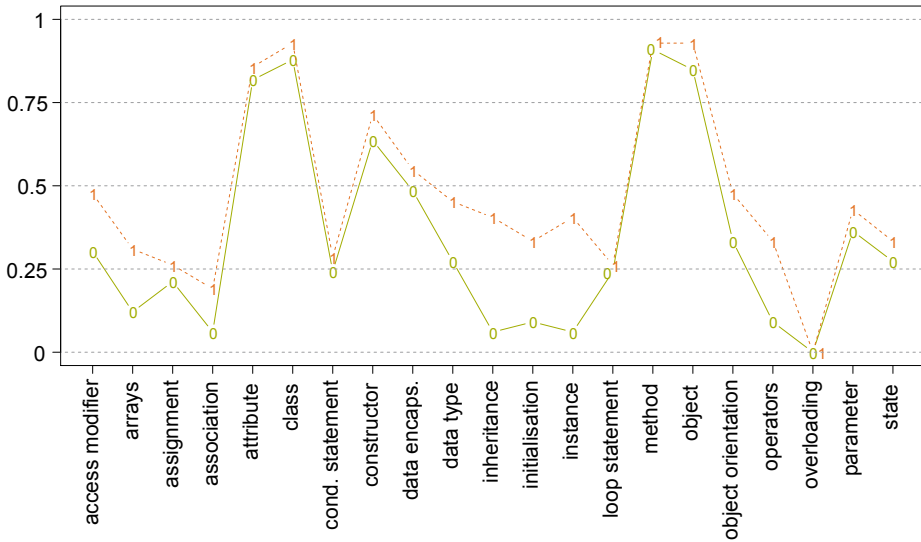
Aggregating over all participants neglects the fact that the prior experience is rather heterogeneous. Fig. 41 shows the development of the knowledge separately for both groups. As can be seen, both groups are gaining relevant conceptual knowledge during the course. Also, there is only very little relevant knowledge present before the course for the group without prior programming experience - as is to be expected, of course. In detail, students with prior programming experience show a better understanding of all concepts after the course except for *conditional statement*, *inheritance*, *instance*, *operators*, and *parameter*, where there is not much difference. Interestingly, they seem to “lose” some knowledge regarding *instance*, *operators*, and *parameters*. However, since the aggregation method is rather coarse, this might for example happen when the students had a very “simple” proposition in the pre map (with a score of 2) and then tried a more complex one in the post map which was incorrect or at least not scored with 2. Additionally, students without prior experience seemingly learned the most about “core” concepts of object orientation: the four concepts with the biggest increase are *attribute*, *class*, *method*, and *object*. The course material put much emphasis on these concepts. Additionally, the “difference” in knowledge between the two groups is decreasing for the post maps. Fig. 51 in the appendix shows the same data but separated between programming experience instead of test. The difference between both lines is plotted in Fig. 52, also in the appendix.

Using the same method as above, new concept landscapes were formed based on concepts that had at least one incident edge with a score of 0. Conversely, these are taken to mean that the students have a misconception regarding this concept. The mean of the columns then, again, is the relative frequency of a misconception occurring for this concept of CL. Fig. 54 and Fig. 56 in the appendix show the results when taking all maps into account and when separating according to test. Fig. 42 shows the results when separating according to programming experience. The difference between both lines is plotted in Fig. 53 in the appendix. As can be seen, the group with prior programming experience shows almost no development between pre and post test. For the group with no prior experience, the misconceptions are almost always increasing in frequency of occurrence, most notably for the concepts *attribute*, *initialization*, and *parameter*. However, even for these the absolute frequency of occurrence is rather low.

The previous aggregations, like in the last case study, were partly based on a manual clustering of the data regarding the attribute “prior programming experience”. It seems worthwhile to additionally use a clustering algorithm in order to find hidden structural aspects in the data that may be masked due to the manual clustering based on an external criterion. For the pre maps, both PAM and MBMM clustering



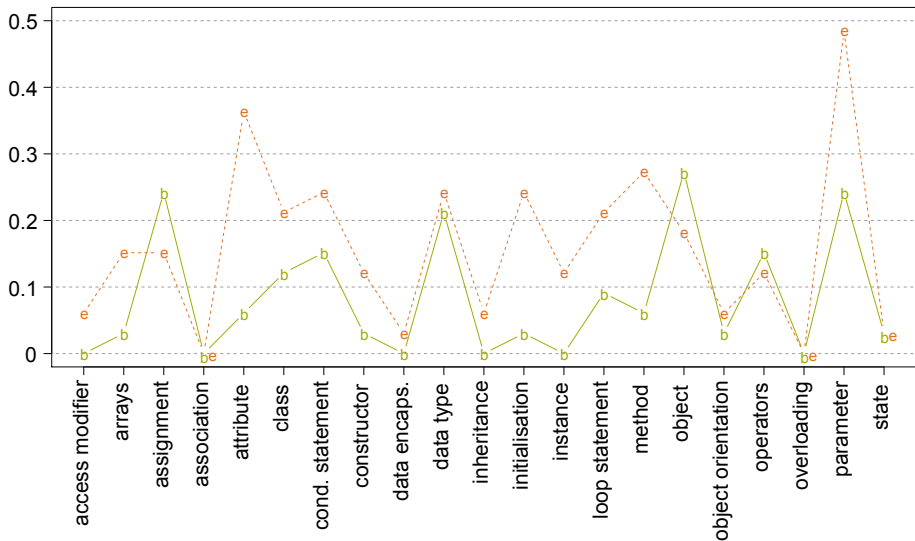
(a) Pre Maps



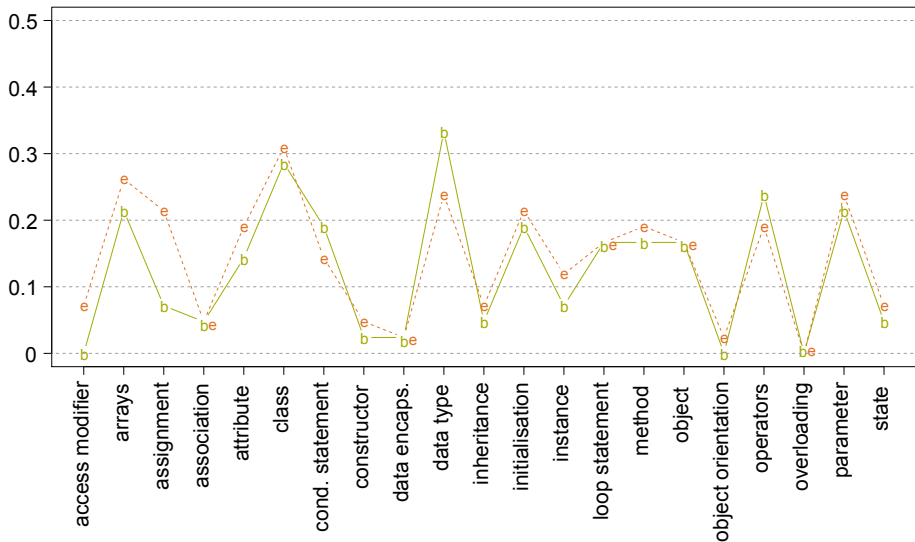
(b) Post Maps

Fig. 41: Development of structural knowledge dependent on prior programming experience. Shown is the percentage of maps that showed at least one “correct” edge (i.e. with a score of 2) incident to a concept (0 = no prior experience, 1 = prior experience).





(a) No experience



(b) Programming experience

Fig. 42: Development of misconceptions dependent on prior programming experience. Shown is the percentage of maps that showed at least one “incorrect” edge (**b**(eginning) = pre test, **e**(nd) = post test).

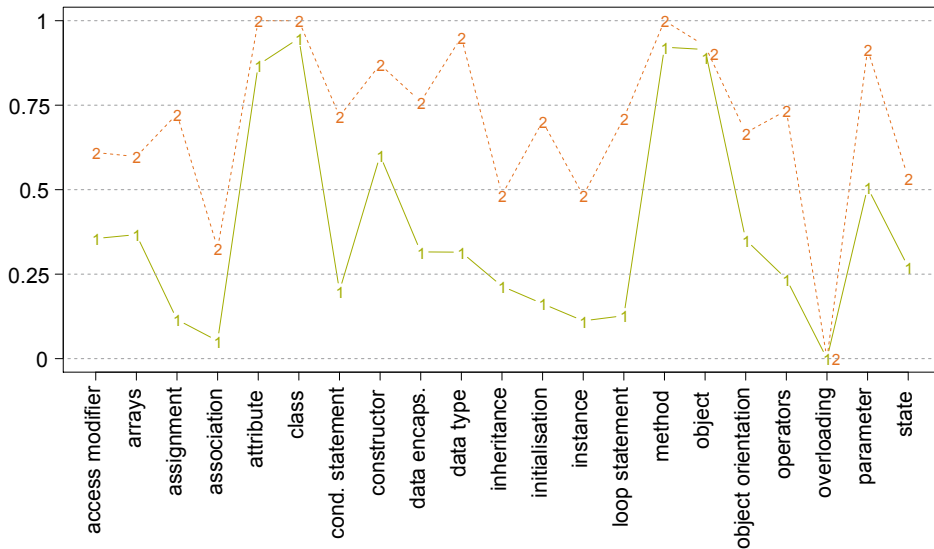


Fig. 43: The probabilities of concept occurrence as identified by the MBMM clustering algorithm for the post maps, shown for both clusters.

mostly identified the empty or nearly empty maps and grouped them together. When using the clustering based on multivariate Bernoulli mixture models for the post maps over all participants, the best model is one with two clusters. Both clusters have about 60% (57.8% and 57.1%) of the maps in common when compared to the manually formed clusters based on prior experience. Also, both clusters are of equal size with 37 and 38 members respectively. So, there does seem to be additional structural information in the data present to allow for another way of forming clusters. Fig. 43 shows the probabilities for each Bernoulli component directly for both clusters. The plot can be interpreted in the same way as the aggregated plots above. The clustering that is found is apparently not sensitive to the (randomly chosen) start values, as the algorithm repeatedly found nearly the same probabilities in 50 runs.

To better identify the characteristics of the clusters, it is interesting to take a look at concepts that have a very high (or low) probability of appearing for each cluster. For a threshold of 0.8, the concepts for the first cluster are: *attribute*, *class*, *method*, *object*. For the second cluster, additionally the concepts of *constructor*, *data type*, and *parameter* are above the threshold. Note that these are predominately concepts that are associated with object orientation. For concepts with a probability less than 0.2, the second cluster only has *overloading*, while the first cluster additionally has: *assignment*, *association*, *initialization*, *instance*, and *loop statement*.

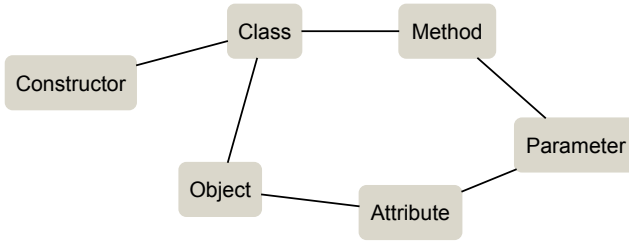
To further investigate the differences between the clusters, the post maps of each cluster were vertically amalgamated by summing edges and a Pathfinder network was created from each landscape separately. To arrive at a small network, a strict pruning was used: For each cluster, only concepts that were at least once “correctly” (score of 2) connected in more than half of the maps were left, all others were removed. Fig. 44 shows the resulting graphs when using the parameters that produce the sparsest results ( $q$  set to one less than the number of remaining concepts and  $r$  set to  $\infty$ ). Clearly, the network of the maps of the second cluster displays a much richer knowledge structure forming an interconnected network of 19 concepts (only *overloading* was pruned). The network structure of the first cluster on the other hand displays only a very limited understanding of the basic concepts of object orientation while nearly all concepts that are more oriented towards programming have been pruned.

Also, like in the last case study, the PAM algorithm was employed as an alternative to MBMM using the graph similarity matrix. The Hopkins index of the data is 0.79 indicating its non-uniformity. The best result according to the  $G1$  index is three clusters of a similar size with 21, 30, and 24 maps respectively. The results indicate structural differences, mostly focusing on the concepts *attribute*, *assignment*, and *object orientation*. The networks are shown in the appendix in Fig. 58 to Fig. 60 for completeness. It seems that the students of cluster 2 seem to have a misconception regarding assignments, since the concept is only connected to *data type*.

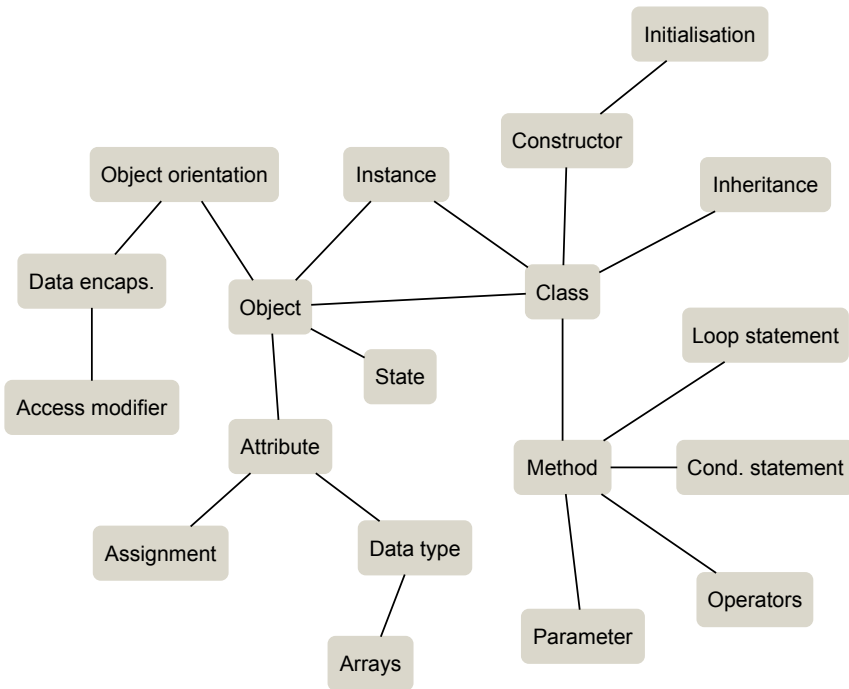
### 12.3.2 RQ2: Connections Between Knowledge and Abilities

The code that the students produced is taken as an indicator of their programming abilities. Analyzing and scoring of object-oriented code is a central topic in educational research ever since OOP has been taught in introductory courses in CS. See e.g. Börstler, Christensen, Bennedsen, Nordström, Kallin Westin, Moström & Caspersen (2008), Sanders & Thomas (2007), or Truong, Roe & Bancroft (2004) for possible approaches and classifications. However, while there are certain approaches to scoring that could be used “out of the box”, the underlying research question in this case needs an analysis method that is closely based on the concepts of CL, as the investigation deals with the connection between structural knowledge and programming abilities.

The methodology used in this case study was as follows: A hierarchy of observable applications (in program code) for each concept of the list was developed in a



(a) Cluster 1



(b) Cluster 2

Fig. 44: Pathfinder networks built from the post maps of the members of each cluster (as shown in Fig. 43). The parameter were set to  $q = n - 1$  and  $r = \infty$ . Before the creation, all concepts were removed that weren't connected in more than half of the maps.

systematic way. For example, an application of the concept *constructor* can be observed in the form of *using* it (i.e. creating an object) or in the form of *creating* it (i.e. defining a constructor). These two applications cover everything that can be observed in a program code concerning the concept *constructor*. Clearly, there are other possibilities of partitioning the “application space” as well, for example by further separating the creation of a single constructor and the creation of multiple constructors. That other possible partitions exist is not important for the scoring method, though. It is only important that there doesn't exist an observable artifact concerning a concept that is not matched by any of the concept's applications. So, for each concept of the list, two experts tried to conceive all possible ways of applying it. The result of actual applications is clearly subjective, however, care was taken that every possible application of a concept was indeed matched by any of the postulated observable applications. Since each of the possible observable applications can be present or absent in a given source code, each of these applications is a testable, dichotomous item.

The only concepts that were excluded right from the beginning were *object orientation*, *class*, *data type*, *constructor*, and *instance*. The first one was excluded because in Java there is object orientation by design. The next two because the use of a modern IDE makes it virtually impossible to distinguish between “implementation by the students” and “implementation by the IDE” due to code completion or auto correct features. Finally, the last two were excluded since they cannot be separated clearly (calling a constructor will also create an instance).

In the end, this method led to 36 items. In the following, an item “being present” means that the observation belonging to that item was actually observed in a code. Some of the items turned out to be trivial, as they were present in all or nearly all programs. For example, for the concept *method* there are the applications of *calling a method* and *defining a method*. However, every student used a method, because they were all printing to the console. Nevertheless, as the code questions were derived systematically and qualitatively, the trivial questions were kept for the analysis as well, for the sake of completeness. Table 12.2 shows all the questions, from now on called CI (code items).

The scoring of the code was done analogously to the concept scoring, i.e. for each concept a binary value should indicate whether or not the concepts were successfully applied in the code. In cases where there are several items belonging to a concept, a decision must be made how to integrate the single values into a value for the concept. In theory, there are three possibilities: Either all of the items of a concept must be present, or a fraction (e.g. the majority), or any item must be present to score a 1 for the corresponding item. In some cases, the items of a

Concept	Level	Item
<b>AM</b>	1	Access modifiers <code>private</code> and <code>protected</code> are used
<b>AR</b>	1	An array with pre-initialization is defined in the code
	2	An array without pre-initialization is defined in the code
	3	An element of an array is accessed
	4	An array is created using <code>new</code>
	5	A method of the class <code>Array</code> is used
<b>AG</b>	1	An assignment is used
<b>AC</b>	1	There is an association between classes present
	2	An association between classes is used
<b>AT</b>	1	An attribute is defined
	2	An attribute of another class is accessed
	3	An attribute of a class is accessed within the class
<b>CS</b>	1	A conditional statement without <code>else</code> is used
	2	A conditional statement with <code>else</code> is used
	3	<code>switch</code> is used
<b>DE</b>	1	The visibility of attributes is <code>private</code> or <code>protected</code>
<b>IN</b>	1	A class is derived from an existing Java class
	2	A class is derived from a user created class
<b>IZ</b>	1	An attribute is initialized
<b>LO</b>	1	A loop is used
<b>ME</b>	1	A method is called
	2	A method is defined
<b>OB</b>	1	A variable or attribute has the type of a user defined class
	2	An object referenced by a variable or attribute is used
	3	An object uses itself with <code>this</code>
<b>OP</b>	1	The assignment operator is used
	2	An arithmetic operator is used
	3	A logical operator is used
<b>OV</b>	1	An overloaded method is called
	2	An overloaded method is defined
<b>PA</b>	1	A method with parameters is called
	2	A method with parameters is defined
	3	A parameter of a user defined method is used in the method
<b>ST</b>	1	The state of an object can be read
	2	The state of an object can be changed
	3	The state of an object impacts program flow

Table 12.2: The dichotomous items used to analyze the code with regard to the application of the concepts of CL. The concepts are given as the abbreviations found in Table 12.1.

concept form a hierarchy in the sense that they are not independent from each other. For example in order to use a parameter of a method (PA3), it has to be declared first (PA2). Or, creating an instance of an array with `new` (AR4) is only possible if either (AR1) or (AR2) are also present. For this analysis, the approach was chosen that a concept is seen as implemented as soon as any of its items are present. This follows most closely the way the conceptual knowledge was analyzed: As long as the student shows any relevant knowledge the concept is included into the analysis of the concept maps. Correspondingly, as long as the student did produce any valid code concerning one of the items of a concept, it suffices as indicator that there is some ability present regarding that concept. Clearly, this measure is a very basic one. However, the scoring of the code using these items is highly reliable and should provide an “overestimation” concerning the validity.

In the end, the analysis should give some insights on the interplay between the conceptual knowledge about a given programming concept and the abilities to actually use this concept in programming. The conceptual knowledge about a concept is externalized in the form of concept maps. More precisely, by all the edges (and their respective scores) that are incident to this concept in the map. The abilities are “externalized” by the source code. Both are mapped into a dichotomous value for this analysis. There are more complex alternatives for both source code and knowledge - however a common scale is needed in every way. Without a theory driven analysis of the knowledge structures of the domain and an analysis of the actual abilities of programming, arguably none of the more complex methods leads to inherently better results. To get an overview over the results of the code scoring, the relative frequency of projects (i.e. the whole of the source code of a participant) in which an item was present can be used. While a complete listing doesn't yield much insight, Table 12.3 does give a summary by condensing the information into percentiles.

Percentile	Code Questions
0% - 20%	IN1, IN2, AR1, CS3, OV2
21% - 40%	AC1, AC2, AR5
41% - 60%	IZ1, AT2, OB3
61% - 80%	DE1, OP2, AR2, AR4, PA2, PA3, OB1, OB2, AM1
81% - 100%	ME1, ME2, AG1, ST1, ST2, ST3, OP1, OP3, AR3, PA1, AT1, AT3, CS1, CS2, OV1, LO1

Table 12.3: The items of CI and the percentile of projects that showed the respective property.

Using the code analysis together with the landscapes of the concept maps presented in the last section, it is now possible to also aggregate the score values for each test (pre or post) and group (prior experience or not) in the same way as above. This way, the relative frequency of “correct” concept occurrences in the concept maps and corresponding observable application in the source code are shown simultaneously. Only the post test is relevant for this comparison. The results for all students are shown in Fig. 45, the results for separated groups are shown in the appendix in Fig. 57, however, there is only little difference between the two groups. Basically, each concept falls into one of three categories:

1. The two values are high and close together. This indicates concepts that were understood well and implemented well, on average. The core concepts of OOP, *attribute (AT)*, *method (ME)*, and *object (OB)* fall into this category.
2. The two values are low and close together. This is the opposite of the first category. *Association (AC)* and *inheritance (IN)* belong to this category, which are the more advanced OOP related concepts.
3. There is a (somewhat large) gap between the two values - this includes all the concepts that are related to procedural programming - *assignment (AG)*, *conditional statement (CS)*, *loop (LO)*, *operator (OP)*, *parameter (PA)*, and *state (ST)* - as well as the more “technical” concepts like *arrays (AR)* and *access modifiers (AM)*, but also *overloading (OV)*.

The only concept that doesn’t really belong to any of the 3 categories is *data encapsulation (DE)*.

When using the results of the clustering from Fig. 43 and comparing the abilities of the students of the two clusters, there are almost no differences to be found (the largest difference between the two groups is 0.14).

## 12.4 Discussion

Concerning the first research question (RQ1) there is a plethora of observations that can be made from the different diagrams. First, and this is positive to note, both the students with and without prior programming experience are actually gaining relevant structural knowledge. So, with only the minimal input that was given to the students they were, on average, still able to develop their mental models of the concepts in question. This holds true for both the students with and without



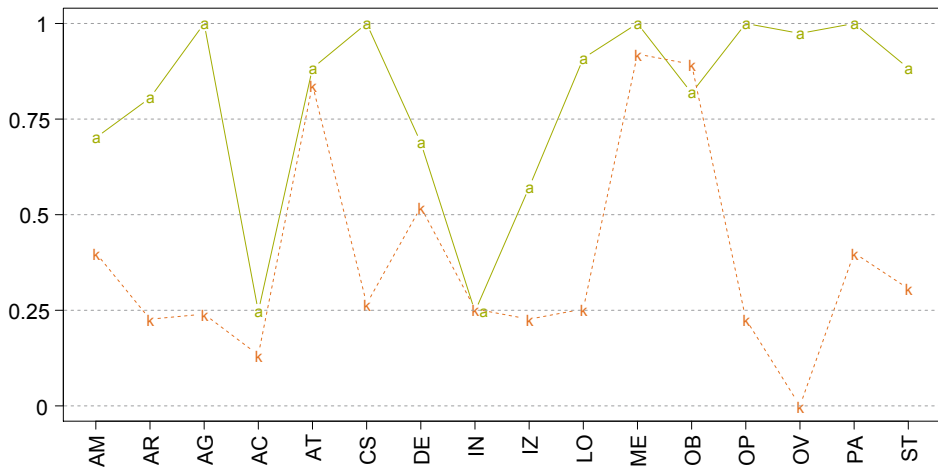


Fig. 45: Difference between conceptual knowledge and programming abilities concerning the concepts of CL (k = **k**nowledge, a = **a**bilities).

prior programming experience. Also, most increase in knowledge is observed for the basic concepts of object orientation (e.g. *class*, *object*, *method*), which were a focus of the worksheets given to the students. The heterogeneity between both groups is visibly reduced which was one of the goals of the course - alleviating the differences between students at the beginning of their CS studies. Looking closer at the differences between the pre maps and post maps, there are some concepts (*inheritance*, *initialization*, *instance*) that students without prior experience show no increase in knowledge in. This could be seen as an artifact of concept mapping, for example if these particular concepts aren't easily incorporated into a map. However, as the other group (with prior experience) made use of the concepts in their maps, this case can rather safely be dismissed. An alternative explanation is this: It seems that those concepts need a certain level of understanding, before students can correctly incorporate them into their mental model. Since the students without prior experience were acquiring this basic knowledge in the course, they were unable to additionally focus on the more advanced concepts. It may also indicate however, that the course material is not suited for explaining advanced concepts to students without any prior programming experience.

Concerning the clusters shown in Fig. 43, the concepts with an inter-cluster difference of more than 0.5 of the corresponding probabilities are (in descending order of the difference): *data type*, *loop statement*, *assignment*, *conditional statement*, and

*initialization*. This list is interesting insofar, as there are no concepts present that are “purely” object-oriented (like *class*). Instead, the list contains the very basic concepts of procedural programming, namely assignment as well as the two control structures conditional- and loop statement. Also, the members of the first cluster tend to focus mostly on the basic concepts of object orientation in their post maps. This may be due to the fact that these concepts were most easily integrated into a map or it maybe due to the fact that these persons didn’t gain enough programming experience in order to form relevant structural knowledge of the concepts more oriented towards programming. The fact that many students did integrate these concepts (e.g. *conditional statement*) into their maps supports the fact that there is a group with only very limited knowledge of the programming concepts. So there are viable mental models that incorporate the concepts and also people who are able (and willing) to externalize these models. However, since both clusters found by the algorithm are somewhat matching the manually formed clusters based on programming experience, it is unlikely that the first cluster just represents students not motivated or not able to externalize more of their structural knowledge in the concept mapping task. Instead, it seems plausible to assume that the clusters are forming a truer picture of the prior experience of the students - the 40% of students who said they had prior programming experience but didn’t end up in the second cluster may simply have overrated their prior knowledge.

Concerning the second research question (RQ2), first there are some observations regarding the percentiles of the code items. Obviously, there are some constructs that will appear in even the most basic Java programs, like the use of an assignment. So, not surprisingly, the corresponding items are present in nearly all projects. Also, even though, for example, overloading can be incorporated in virtually every program, not every programming task lends itself for e.g. the creation of a hierarchy of classes. Since the programming tasks were rather small - in order to make it possible for the students to finish the tasks in the two and a half days - it must be assumed that these items are not necessarily only non-present in most projects because they are more difficult than the rest, but simply because there was no clear necessity to use the corresponding constructs. Therefore, most insight can be gained from items that are present at least in some, but not in all of the projects.

The results (Fig. 45) show that abilities do not strictly follow knowledge. Instead, three categories of the relationship can be identified. There is a group that is understood and applied well, a group that is only applied well and a group that is neither understood nor applied well. It is interesting to note that there are concepts of object orientation in both of the first categories. This clearly shows, that several of those concepts (like *inheritance*) are seemingly harder to grasp than the rest, while

students actually were able to understand *and* apply the basic concepts of object orientation (e.g. *object*) after the course. Next is the huge difference between the representation of knowledge and the usage in the code of the 3rd group of concepts like *loop*. This particular concept might have been hard to integrate in the concept maps, since it is part of a few concepts dealing more with procedural programming than with object orientation. But it may just as well show, that “understanding” those concepts is not a trivial task and takes considerably more time than learning how to apply them. This is especially true for overloading, where the high value for application mostly comes from calling an overloaded method. Together, this indicates the difficulty inherent in learning (object-oriented) programming: There are several groups of concepts that are all needed to create a “real” OO program, but those groups of concepts are showing radically different results even for those students that had prior programming experience! Finally, there is an interesting observation on two very similar concepts. *State* on the one hand and *attribute* on the other. The state of an object is defined by the value of its attributes. Looking at the values for the conceptual knowledge of both concepts, however, there is a big difference, which doesn’t hold true for the application, where the results for state and attribute are close together. So *state* is a concept that is clearly being used by the students but not understood. It is not surprising that the students don’t fully grasp the concept of state transitions of objects after this particular introductory course, though.

The fact that there is only little difference between the abilities of the two groups and between the persons of the two clusters identified above indicates that overall the scoring method used does not offer very much spread over the projects. However, it also indicates the success of the courses, as one of the goals was that every participant has a working project in the end - this alone indicates the presence of many items of CI.

In conclusion the results show that there are seemingly different kinds of OOP-related concepts. Some are basic enough for students to learn on their own in a rather short time frame, others require a certain level of prior knowledge in order for students to be able to incorporate them into their mental model meaningfully. However, it is possible for students to learn about application of certain concepts without fully understanding the underlying concepts. In a way, this can be seen as supporting the idea of the taxonomy given by Fuller et al. (2007). It suggests that especially for computer science the cognitive processes of learning objectives can be separated in two (independent) dimensions of “Producing” and “Interpreting”. Also, the results show that it is possible to have a rather heterogeneous group of students working on the same material and still have learning opportunities on both

groups. The students that had already programmed focused on certain concepts in their learning and simply created more complex programs. The students without previous knowledge picked up basic elements of OOP while ignoring (or not fully understanding) the more advanced concepts.

Concerning the analysis methods used, concept landscapes and the way they were used in this study proved to be effective. Even though there was lots of information lost in the way the maps were aggregated, it still allowed an insight into the structural development of students' knowledge. The clustering worked well in identifying two groups that differ from the manual selection and that possess a visibly differing richness of their knowledge structures. Pathfinder networks were helpful in making this distinction clear. However, especially concerning the analysis of source code, there is still a lot of room for improvement. While the system of using code items worked reasonably well concerning reliability, there is no guarantee concerning the validity of the measurements.

## **Part V**

---

### **Conclusion**



## 13 | Summary

This thesis - in the last two parts - has presented the notion of concept landscapes both in theory and practice. As chapter 6, based on chapters 3 and 4 has established, concept mapping has proven to be an effective method for teaching, learning, and assessing of conceptual knowledge - especially when focusing on the structural aspects of concept interdependence. Concept mapping is fundamentally based on the theories of meaningful learning and Constructivism, which have both been presented in chapter 3. Also, the psychological definition of a concept as mental representation together with the psychological function of integration form a basis on which concept mapping can be better understood. The validity and reliability of concept mapping assessment tasks has been established by literature, albeit not beyond any doubt, especially when considering the plethora of different concept mapping tasks to be found in literature. This thesis therefore has focused on the “gold standard” concept mapping task of creating a map from scratch with the only restrictions being a list of concepts or a restricting list of concepts. The most severe limitations of concept mapping from an epistemological point of view is the restriction to propositions between two concepts at most, which is not enough for expressing arbitrary facts. From an educational point of view, it is possible to assess learning objectives that cover almost the entire “Interpreting” dimension of the taxonomy by Fuller et al. (2007), as reasoned in chapter 6. Even though concept mapping falls short in assessing skills or competencies of a person, both are typically based on a cognitive component that may be assessed by concept maps.

Based on these foundations, a novel view on the application of concept mapping in investigating the state and development of structural knowledge has been presented in chapter 7. Instead of focusing solely on the measurement of a single person, or many measurements of single persons analyzed in isolation, the data of a group of persons is aggregated. Inspired by the ideas of data mining, this aggregated data can be used to gain new insights. While motivational factors, the fluency regarding concept mapping itself, the specific location and time of the concept mapping task and other variables will, whether detectable or not, influence the results of each concept map, many of these influences can be expected to cancel out by aggregation. There are two fundamental ways in which concept maps of different persons can be aggregated, as shown in Fig. 11. A concept landscape can be horizontal - focusing on the development of knowledge during time - or vertical - focusing on the state of knowledge at a single point in time. In practical settings, often a combination of both of these aspects will be used. The method of aggregation, as shown in Fig. 12 can either be a “loose” accumulation of maps,

that remain identifiable in the aggregated data, or it can be a more transformative amalgamation that results in a new graph which is influenced by the constituent maps of the landscape.

Concept landscapes are only an abstract notion of a set of data items. In practical settings it is the analysis and the actual method of aggregation that determines the usefulness of the approach. This thesis presented two basic analysis methods. Cluster analysis is best suited to identify inherent differences within an aggregation and typically works with vertical accumulations. While many clustering algorithms and similarity measures can be used, a latent model based approach based on multivariate Bernoulli mixture models and partitioning method using, for example, graph similarities as measure of distance have been presented and shown to work well on actual data. Creating Pathfinder networks from concept landscapes is a way of pruning edges that allows discovery of salient structural information. Therefore, an amalgamation is the chosen method of aggregation for Pathfinder analysis in order to identify the common structural elements in the data. Pathfinder networks have been chosen over other methods of graph pruning or dimensionality reduction, since graphs are a natural model for concept landscapes and Pathfinder networks originated from the use in analyzing structural knowledge. These two basic approaches will usually be accompanied by further ways of analyzing the results: Several graph measures were presented as well as the analysis of frequently occurring subgraph structures. These are focusing on the micro structure of graphs. Finally, visualizing concept landscapes often offers insights into the structure of the aggregated map data “at a glance” and also eases the interpretation of Pathfinder networks. The development of knowledge can be displayed very well using the visualization of a horizontal accumulation.

Since data mining approaches are focusing on large amounts of data and all sub tasks in the context of concept landscapes can be supported well by software solutions, chapter 8 presented a tool-chain for working with concept landscapes in computers, as shown in Fig. 20. At the beginning of a study, concept maps must be drawn or collected. The online editor CoMapEd allows researchers to conduct surveys with flexible settings while also offering functions for the participants, like automatic, online saving of their current work and the possibility of exporting the concept maps. In the subsequent analysis the techniques presented in chapter 7 are all best suited for computer aided analysis. The R package CoMaTo provides this support by offering implementations of concept landscapes as data structures as well as all analysis techniques presented in this thesis. Additionally, the analysis of the (text) material used in an educational process is a valuable resource for the interpretation of results, by providing pointers to problem spots in the material or to



particular artifacts in the concept landscapes of learners. ConEx provides some basic features that guide the analysis process by offering extraction of frequently occurring or salient nouns/concepts and the sentences/propositions that contain them.

Finally, the previous part presented several case studies in which concept landscapes and the associated analysis techniques were applied in real world scenarios with actual research questions. The results concerning computer science education will be discussed in detail in the next chapter. From the perspective of concept landscapes, the studies have shown that in particular forming Pathfinder networks from sets of concept maps is a fruitful way of arriving at interpretable results. The development of knowledge along the course of a lecture has been visualized and it has been shown that misconceptions are developing alongside of “correct” knowledge. Also, the average knowledge structure of beginning CS students has been analyzed and illustrated with the help of a vertical amalgamation of concept maps. The attendance of the newly introduced compulsory school subject Informatics in Bavaria has a measurable impact on the knowledge structures of beginning CS students. Finally, the interdependence of conceptual knowledge and programming abilities has been investigated. It has been shown, that conceptual knowledge develops even if only minimal theoretical input is provided to students - as long as they are able to practically apply the relevant concepts in programming tasks.



## 14 | Discussion

To relate back to the introduction in the very first part, this thesis has been centered around the research area “student understanding” that was identified for research in computer science education - in particular the “investigation of students’ mental and conceptual models, their perceptions and misconceptions” (Fincher & Petre 2004, p. 3). This in turn is relevant in order to “develop models that can explain how educational process take place [...] and analyze strategies for intervention” (Klieme et al. 2008, p. 3), which has been defined as one of the goals of modern educational research. This thesis presented analysis methods based on educational data mining, related to the general goals of “clustering”, “relationship mining” and “distillation of data for human judgment” (Baker & Yacef 2009, p. 9) - as presented in the three cases studies. It can also be used to analyze interventions - as shown in the third case study - but more generally investigates mental models of students. This chapter will discuss what has been presented in both the third and the fourth part separately. In the same vein as Fig. 2 shows the interconnections of several areas of research that provide the basis for this thesis, Fig. 46 shows the results, their interconnections, and how they affect the research areas that provided the basis. Aside from the case studies, the two specific research questions of this thesis have been presented in chapter 2:

1. *How can methods of data mining be applied to sets of concept maps in order to identify common elements and differences between the individual maps?*
2. *How can software support the workflow of the research design presented in Fig. 1?*

The first question has been answered by the findings of chapter 6 and chapter 7: Concept landscapes can be used together with data mining approaches to identify both common elements and the differences between individual concept maps. In particular, cluster analysis, Pathfinder networks, graph measures, and visualization techniques have been presented for the task. The case studies have shown the successful application. The second question has been answered by chapter 8, where software for each of the pivotal points of the research design of Fig. 1 has been presented, as shown in Fig. 20. Specifically, the computer based drawing and collecting of concept maps, the automatic analysis of textual input data, and the computer based analysis using the methods of chapter 7 have been made possible by the three software projects CoMapEd, ConEx, and CoMaTo.

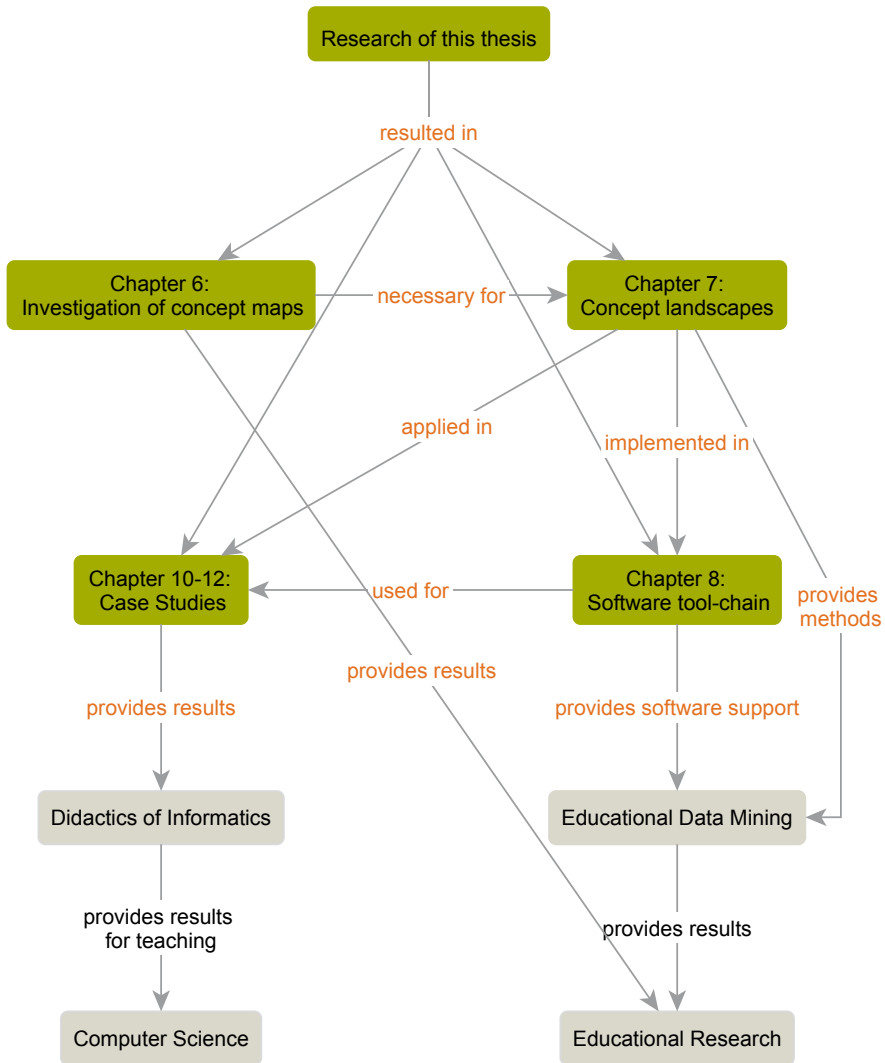


Fig. 46: The results of this thesis and how it affects the different research areas.

Concept maps have been chosen as the method for externalization because of their established validity and reliability, the existing psychological and educational theory behind it, and the additional benefit they can offer to participants. On the downside, however, the method of concept mapping has to be explained to participants and may even require some form of training until they become fluent enough in the process. Also, the relative freedom of expression, especially for pen and paper based maps, often leads to highly variable results concerning the actual syntax and semantics of concept maps. Therefore, alternative methods of externalization may present viable alternatives for the specific task of research on knowledge structures. As has been noted in chapter 6, relatedness judgments may provide more accurate results concerning the structural organization of concepts. Among other factors, the non-existing benefit for participants and the influence on motivation must then be considered though. It remains questionable, whether a large scale study with useful results can be obtained from using relatedness judgments. This holds true especially since it is far easier to, for example, detect a nonsensical concept map of a student not interested in taking part in a study, than it is to detect relatedness judgments that were simply guessed or randomly chosen. Concept mapping has proven to be a working and established choice - however investigating different forms of eliciting structural knowledge may provide more insights both into the workings of concept landscapes and into the strengths and weaknesses of concept mapping.

The four ways of aggregating concept map data into concept landscapes are not covering all combinatorial possibilities of aggregating a set of maps. However, longitudinal and cross-sectional studies are possible, which cover typical research scenarios. Forming either an amalgamation or an accumulation does cover all possibilities as either the concept maps remain identifiable as entities in the aggregation or not. The formal definitions that were given in chapter 7 serve the purpose of clarification and precise definition but are neither the only, nor the single right way of modeling the four types of landscapes. The two central methods of clustering and Pathfinder networks are able to achieve useful results. Nevertheless they must always be seen as an exemplary selection of all the possible ways of analyzing the data of aggregated concept maps. There are other ways of pruning graphs, other similarity measures, or ways of arriving at a distance matrix for clustering and also other ways of defining latent class models for concept map data. There is no particular reason for using the centrality of a graph, for example, but not some other graph measure - except that there is an explainable expected connection between the graph measure and properties of structural knowledge. The goal of this thesis is to present a working analysis method that allows investigations into knowledge structures that are based on a thorough theoretical basis. Other ways

of investigating concept landscapes are not dismissed by this choice, though, and should be considered.

What has been hinted at in chapter 7 and followed stringently in the case studies is, that typically neither a purely quantitative nor a purely qualitative analysis yields optimal results in exploratory, fundamental educational research. Instead, both approaches have been used simultaneously and - one might say - non dogmatically in order to improve on each other. For instance, finding clusters and then using a t-test to identify that one cluster possesses denser maps is a quantitative approach. Then continuing by searching graph communities in these clusters and inspecting them regarding their intersections and differences, based on the given subject matter context, is a qualitative analysis of the data. By using the strengths of both methods, a much broader variety of insights into the data of concept landscapes can be gained.

The software tool-chain that was developed in the course of this work has proven to work well in research settings. A choice was made in each of the three scenarios to develop software instead of using an existing solution due to the restrictions that these solutions presented and the judgment of how costly an implementation is. Nevertheless, for each project several design choices were made, none of which was without alternatives. This is especially relevant for the drawing tool CoMapEd, since this is the one that most people interact with in actual research settings. Therefore, this software must provide the necessary ease of use and functionality that participants need in order to keep up their motivation. The choice to implement it based on a browser is certainly valid given the variety of settings encountered in actual studies and it has been shown to work well. Other choices however, like the particular format of exporting, or the actual methods of interacting with the drawing area, may certainly have to be reconsidered given actual user feedback and might change in the future. The analysis with CoMaTo based on R makes sense, given the flexibility of the approach. The analyses of the case studies have shown, that it is a valuable support for working with concept landscapes. Improvement are therefore mainly in the form of implementing new features, as briefly described in the next chapter. ConEx so far can only be seen as an experimental study. Many different existing methods of extracting salient words from texts exist and should be considered for the problem at hand.

## 14.1 Case Studies

The case studies are providing the research results in the form of pedagogical content knowledge for subject-matter didactics. The particular set of studies mostly presented insights into the effects of teaching and learning central, basic concepts of computer science - often related to object-oriented programming. Also, the participants were always students at the very beginning or in the first semesters of their studies at university. While these specific settings are very interesting as they allow insights both into secondary education as well as education at universities, clearly, the scope of studies can and should be broadened. This encompasses studies with more experienced students as well as different, more specialized fields of computer science. Nevertheless, the basic concepts investigated in this work are very important, as they form the foundation on which the rest of computer science education at the university builds. When considering the implications of meaningful learning, conceptual change and Constructivism, it becomes very hard if not impossible to develop well formed, densely connected conceptual knowledge and skills without a well-formed foundation.

Concerning the method of conducting the studies, the computer based collection of concept maps is clearly preferable to the pen and paper methods. The number of participants in the second case study, for example, presented an obstacle to analysis that could only be solved by paying students to digitalize the maps. Software based collection would have made this step unnecessary and the study cheaper. Also, when using software, a restricting list of concepts is far more convenient to present to the participants as it reduces ambiguities due to typing errors or due to using different forms (like plural) of the concepts given on the list. Additionally, when collecting concept maps with pen and paper and then manually digitalizing them afterwards, a new source of error is introduced into the process the severity of which is difficult to quantify.

The first case study has provided insight into the development of knowledge and misconceptions, It has shown that learning, as observed, is not a straight forward process - even when taking into account a whole group of learners. So far, it remains unclear whether this process is natural in the formation of personal mental models or a specific artifact of the non-majors that might struggle more with the CS concepts than students who willingly chose computer science as their field of study. Nevertheless, it seems worthwhile to evaluate how teaching can be improved and how students can be made aware of their misconceptions easier and if possible, redeem them early on. This could, for example, make explicit use of the concept landscapes that were formed by the students' concept maps. After all, concept

maps are an established teaching aid as well and making misconceptions explicit is one of the fundamental ideas of conceptual change.

The second case study has presented a method of analysis that can be used to investigate prior knowledge of a group of learners. The structural information is especially valuable since it can be used to present topics in a way that fosters meaningful learning more easily by taking into account what concepts might already be present in the knowledge structure and adapt teaching accordingly. Clearly, the information extracted from the concept landscape doesn't hold for each individual, but teaching to a larger group of learners always requires certain compromises and taking into account "general" prior knowledge is a reasonable approach, as long as the idiosyncrasies of learners are not neglected. The identification of different groups by clustering according to graph similarity has shown interesting insights, especially when taking into account the other (personal) information provided in the survey.

Finally, the third case study has shown that it is possible to introduce basics of object-oriented programming with only minimal input and practical applications and still have conceptual knowledge develop. Clearly, this approach also fosters the creation of misconceptions, but taking the first study into account it seems doubtful whether this can be avoided at all, given a reasonably complex subject area. The interconnections between abilities and knowledge are more diverse than might be expected initially. This may, to some degree, be influenced by the actual analysis methods used in the case study, especially the scoring of the source code. However, it is also an indicator - to some degree - for an independence between "knowing" and "doing" in OOP, which has been postulated by the taxonomy of Fuller et al. (2007) for computer science. It is certainly worthwhile to investigate this dependence further with other studies.

In conclusion, the application of concept landscapes has proven to be useful in practical settings, as the case studies of the last part have shown various interesting results concerning the knowledge development in computer science education. Having a working software tool-chain available opens the door for an economic and flexible way of monitoring certain aspects of educational processes in exploratory settings with minimal prior work. Educators or researchers who are interested in the development of students' knowledge can incorporate concept mapping into their teaching activities, collect, and analyze the maps electronically with minimal additional overhead incurred. The case studies, which have all been exploratory in nature, can serve as a blueprint for research questions, subsequent analysis methods, and interpretations. Since currently the established knowledge about learning in computer science is not yet very far developed, exploratory case studies



---

are needed as the basis for further research. Using these results to develop models of learning that can then be tested is one of the next steps towards better (research in) computer science education.

“Psychologists have amassed a large amount of empirical research on various factors that impact the ease of learning and transferring conceptual knowledge. [...] Putting these suggestions to used in class-rooms [...] could have a substantial positive impact on pedagogy” (Goldstone & Kersten 2003, p. 616).



## 15 | Further research

The work presented in this thesis has been rather broad. Necessarily, single aspects have only been investigated in limited depths. This chapter outlines some of the aspects that deserve further inquiry beyond the work presented here.

First, concerning the concept landscapes and their analysis methods, a more thorough investigation of the mathematical or statistical properties of the analysis methods and the results generated therefrom should be carried out. This includes, for example, tests that indicate the suitability of given methods or that measure to which degree the results are actually significant for the set of maps used. This could eliminate some of the guesswork in the interpretations still necessary now when analyzing concept landscapes. Also, the analysis methods themselves can be refined further. Especially concerning the visualization of horizontal landscapes there is a much broader variety of possibilities than alluded to in the preceding chapters. The results of data visualization techniques in other fields of research must be taken into account in order to judge what might be beneficial for concept landscapes.

Next, the software that was presented in chapter 8 has several possibilities for improvement. For CoMapEd the inclusion of some form of recommender system for the edge labels is an interesting aspect of further development. If users see possible suggestions for labels as they type and the list of labels is filled with meaningful examples, they might choose these suggestions. This would make automated analysis of edge labels far easier, as the semantic meaning of the suggestions could be predefined. Also, it would not unnecessarily restrict users, as they could still choose to type their own label, if they desire. Whether this leads to “better” concept maps or just caters to the laziness of the drawing persons and therefore probably diminishes the validity of the externalization process remains to be seen of course. CoMaTo can be expanded by further aggregation methods, import and export formats, and analysis techniques. ConEx so far is only of experimental value and can clearly be improved much further. The inclusion of more languages, more elaborate methods of extracting terms and better integration into the rest of the tool-chain are examples for further development steps.

Finally, concerning the case studies, there are several interesting opportunities for gaining deeper insight into the knowledge development in computer science - as has been noted in the last chapter already. First, a group of experts should be investigated in order to have a more clearly defined baseline of what is considered to be a “good” structured knowledge of computer science. Next, the development of

knowledge should be more closely related with the development of actual abilities as, clearly, the trend in assessments seems to favor the latter. Finally, areas of CS in which a rich framework of conceptual knowledge is necessary may provide a fruitful opportunity for research with concept landscapes. Examples of such areas are software engineering, algorithms or also areas like operating systems or networking in which students are often required to “know” more than to “do”.

The next two sections present aspects of further research that go beyond the research of this thesis but instead provide directions that may be worthwhile investigating with concept landscapes.

## 15.1 Identifying Threshold Concepts of CS

Threshold concepts as presented in the chapter 3 are an important aspect to consider for teaching in any subject. Concept landscapes might be employed to identify these concepts for computer science. To this end, a longitudinal study would have to be done where a group of learners are continually working on a concept map. These maps could then be accumulated horizontally. Following the characteristics given in section 3.2.3, it seems reasonable to expect the following observable structural attributes in horizontal aggregations for threshold concepts:

**Transformative** Once the concept appears, major restructuring of the map is expected to happen.

**Irreversible** Once the concept appears, it stays in the map.

**Integrative** The concept appears not “at the fringe” of the map, but is connected more centrally right from the beginning.

**Bounded** The concept appears in a central position in the maps of experts.

**(Potentially) troublesome** There is a number of maps in which the concepts does not appear or appears much later then it was presented to the learners, for example.

In a previous attempt to identify such concepts in the context of computer science, Eckerdal, McCartney, Moström, Ratcliffe, Sanders & Zander (2006), in reference to the *fundamental ideas of computer science* (Schwill 1994), suggest *abstraction* and *object orientation* as threshold concepts. Additional candidates are, for example, recursion as the second case study already indicates or the theoretical concepts of

computability and tractability. Also Edmondson (2005, p. 24) describes the potential of concept maps for identifying (based on manual observation) “essential critical nodes” that resemble the idea of threshold concepts.

## 15.2 Combining Knowledge Space Theory and Concept Maps

The theory of knowledge spaces originally described by Doignon & Falmagne (1985) is a mathematical model of the possible “knowledge states” of a given set of “facts” (or items that can be used in assessment) that can be learned (or solved). The knowledge space is a subset of the power set of these facts. The reasoning is, that there may be dependencies between the facts such that one fact must always be learned before another fact, for example. Therefore, not all possible combinations of facts will exist in the real world and the knowledge space reflects that. Identifying how the knowledge space actually is structured is a difficult process, however, and mostly done by querying experts. Albert & Steiner (2005) suggest using knowledge space theory for the validation of concept maps.

Concept landscapes offer an alternative approach to the problem of knowledge spaces and additionally, knowledge spaces offer a way of modeling the structural knowledge externalized in concept maps. It may be possible to identify the dependencies of knowledge about concepts by analyzing how and if or when they appear in a concept landscape most probably by using vertical aggregations of several points in time. It might be possible to extract the relations between states of the knowledge space by statistically analyzing the concept landscape. Knowledge about the actual structure of the knowledge space in turn allows identifying problem areas in instruction - for example concerning the ordering of topics.



## 16 | Acknowledgements

Writing this thesis over the last three and a half years has been an interesting and fun - but also at times very stressful - venture. To see the project develop over time and now, finally, its completion drawing near, is a unique emotional experience. While writing the thesis has been a personal endeavor, first and foremost, as John Donne is quoted: "No man is an island". Consequently, arriving at this point would not have been possible without the assistance and guidance of other persons.

The idea of investigating concept maps originally has been suggested to me by my supervisor Prof. Dr. Peter Hubwieser. He has also been actively engaged in the analysis of the first case study and has been the driving force behind the AVIUS project that yielded the data for the second case study. Besides this very specific help, he has also been guiding the process of writing this thesis right from the beginning. For all of this - and the possibility to write this thesis in the first place - I remain thankful.

Also, I'd like to thank the professors Gudrun Klinker, Ph.D., Christopher Hundhausen, Ph.D., and Dr. Johannes Magenheim for agreeing to be part of the examination committee of this thesis.

It would not have been possible to collect the large amounts of data without the help of all the other people, students as well as researchers, at the working group "Didaktik der Informatik". In particular Marc Berges, who collected the data of the third case study and was also actively engaged in the analysis of the corresponding original research study - especially concerning the analysis of the source code. Also, he helped with collecting the data of the AVIUS project.

The original manuscript was proof-read by Julia Kurth and the cover of this print was designed by Peter Wagner.

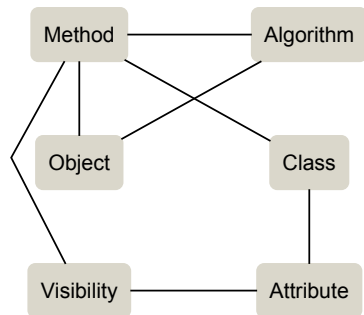
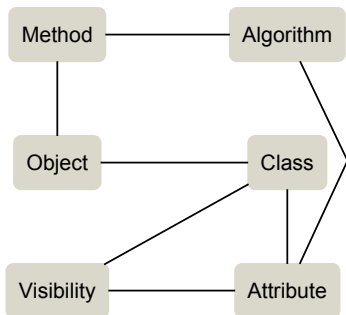
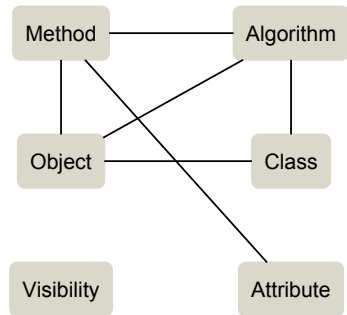
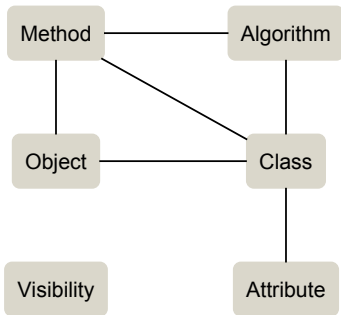
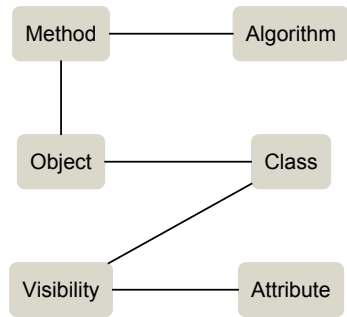
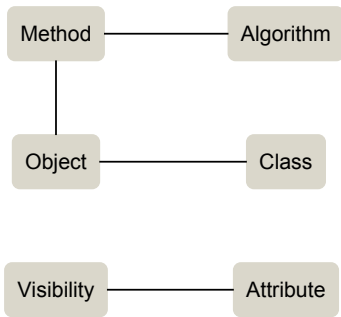
Finally, there have been many other persons who supported me on a personal level during the process of writing this thesis and whom I have to thank - encompassing, in particular, my parents.

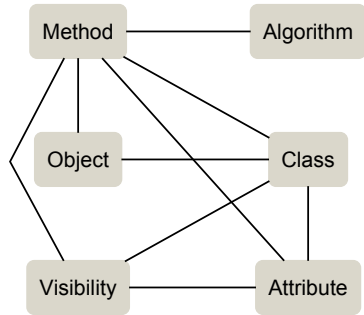
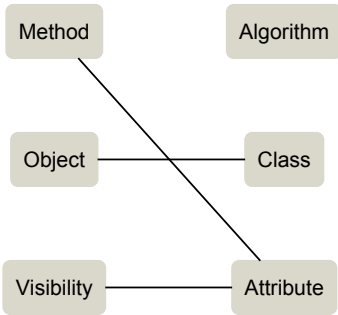
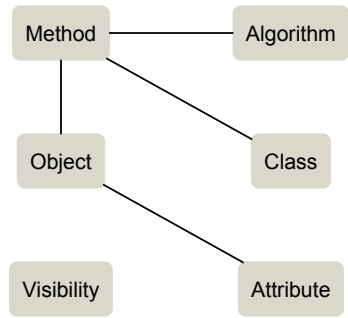
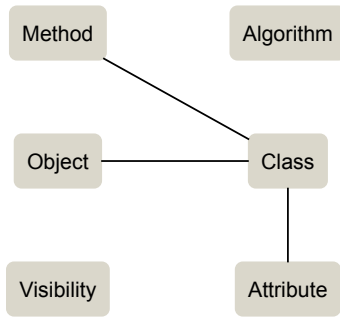




# A | Example Maps for Pathfinder Analysis

The following ten concept maps have been used as the basis of the Pathfinder example given in section 7.2.2.1. Since only the structure of the maps is relevant for the result, no edge labels are given.





## B | Additional Diagrams for CS1

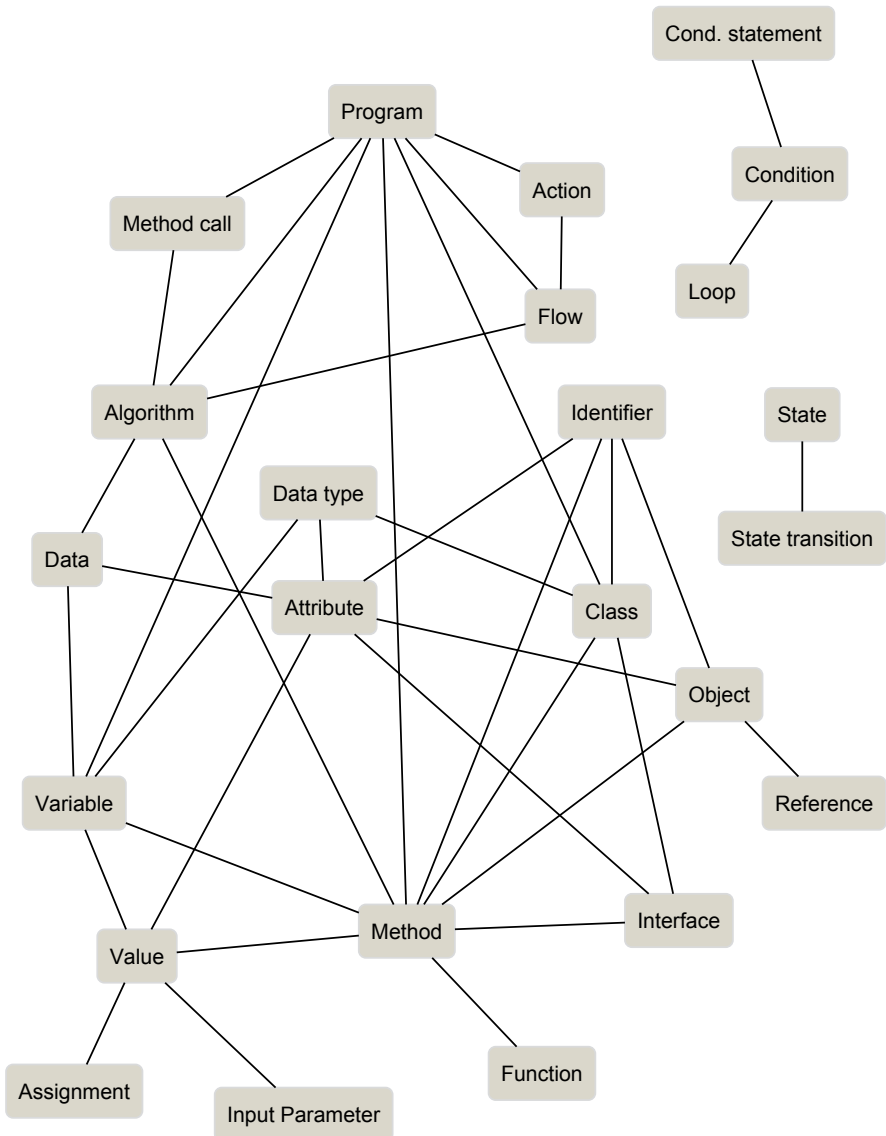


Fig. 47: The second expert map containing all the propositions that were presented between MT1 and MT2.

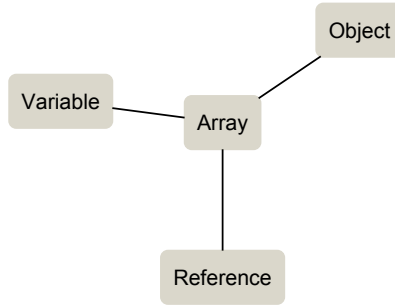


Fig. 48: The third expert map containing all the propositions that were presented between MT2 and the exam.

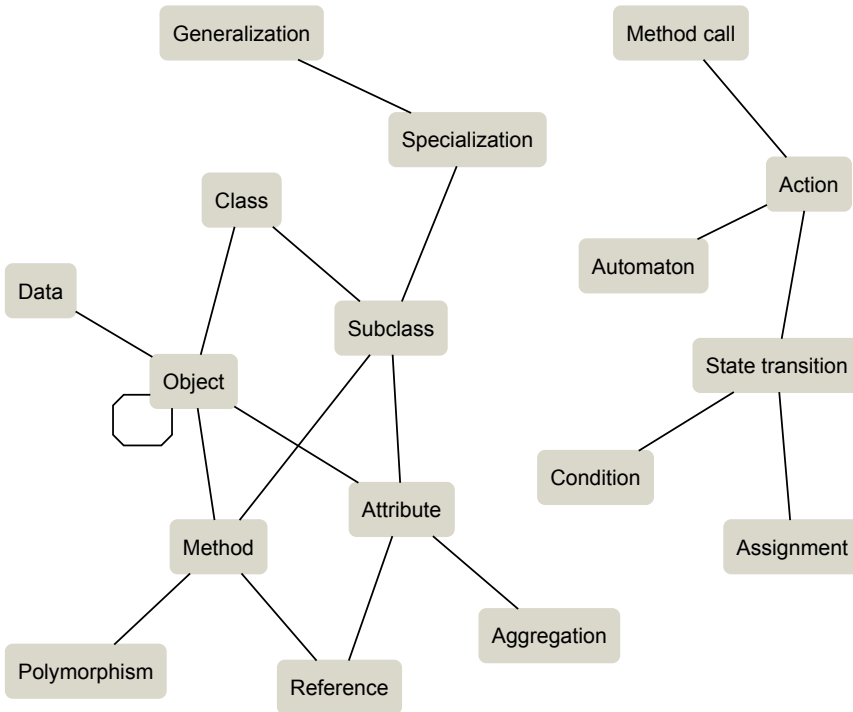


Fig. 49: The fourth expert map containing all the propositions that were presented between the exam and POST.

## C | Additional Diagrams for CS3

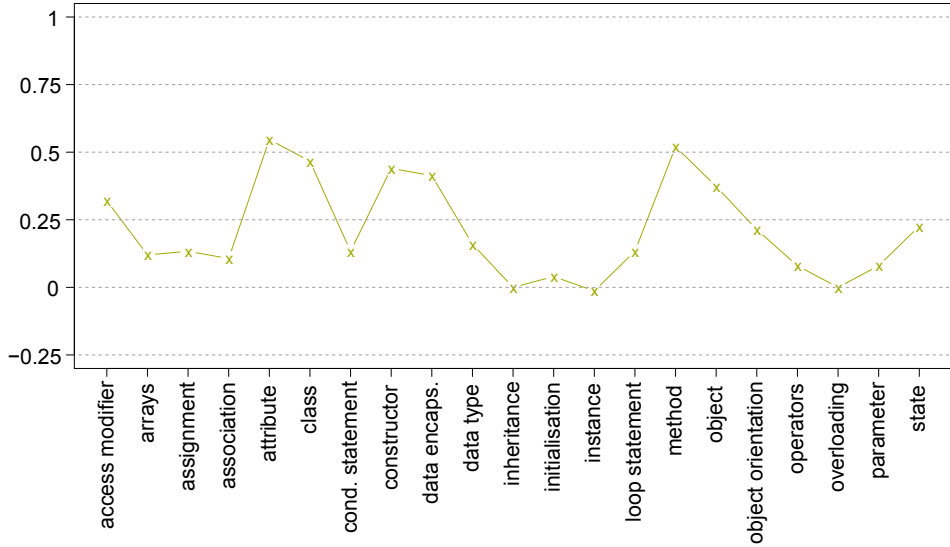
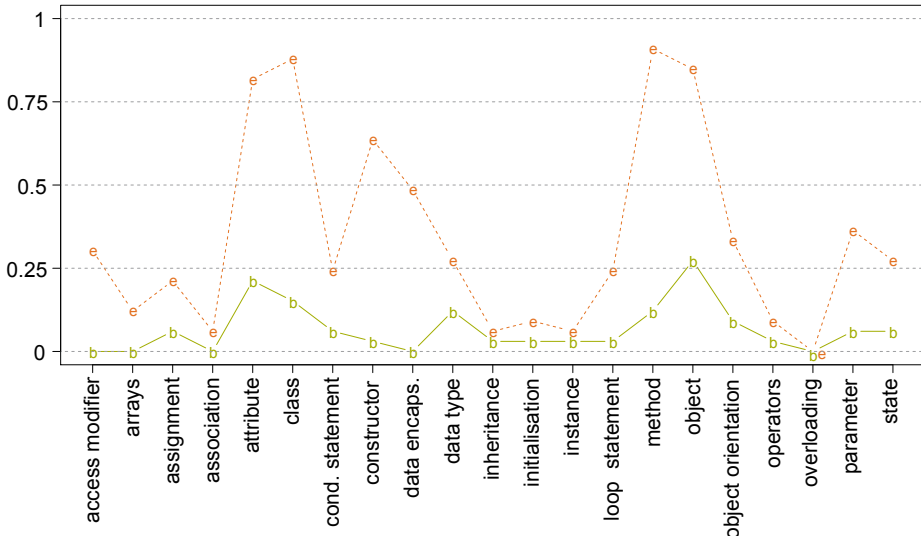
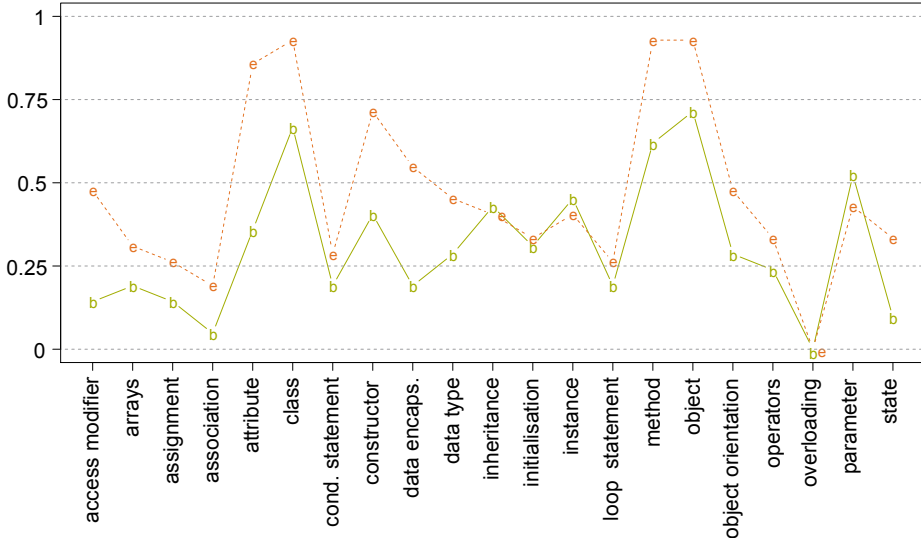


Fig. 50: Development of relevant structural knowledge over all participants. Shown is difference between post and pre test of the diagram in Fig. 40.

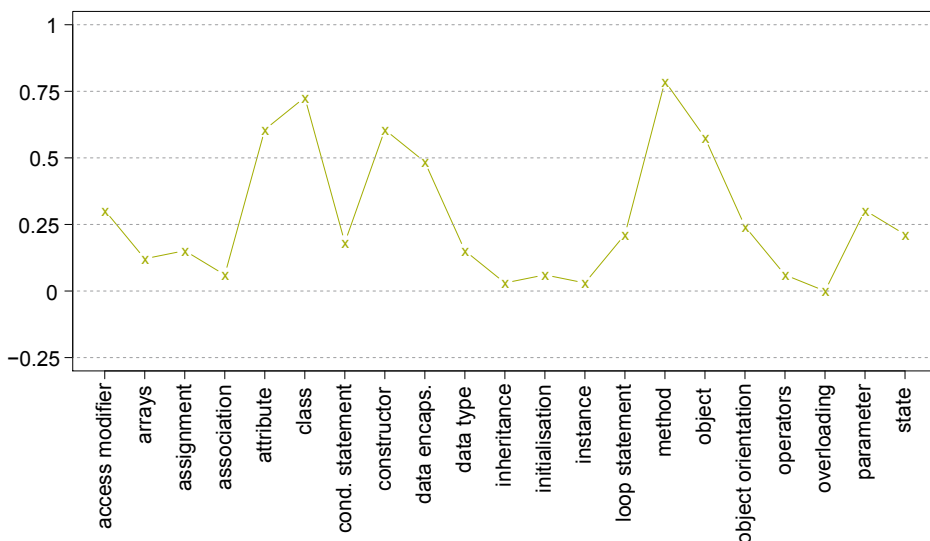


(a) No experience

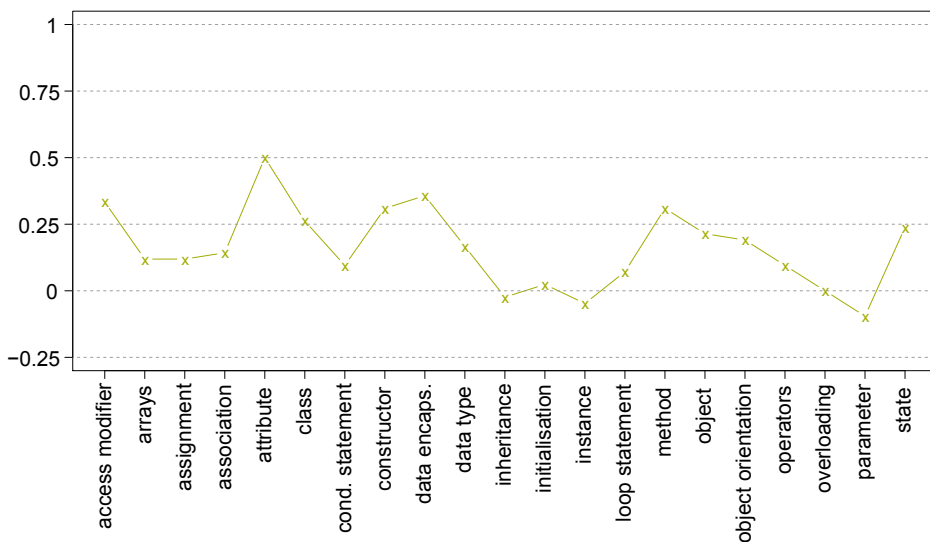


(b) Programming experience

Fig. 51: Development of structural knowledge dependent on prior programming experience. Shown is the percentage of maps that showed at least one “correct” edge (i.e. with a score of 2) incident to a concept (**b**(eginning) = pre test, **e**(nd) = post test).

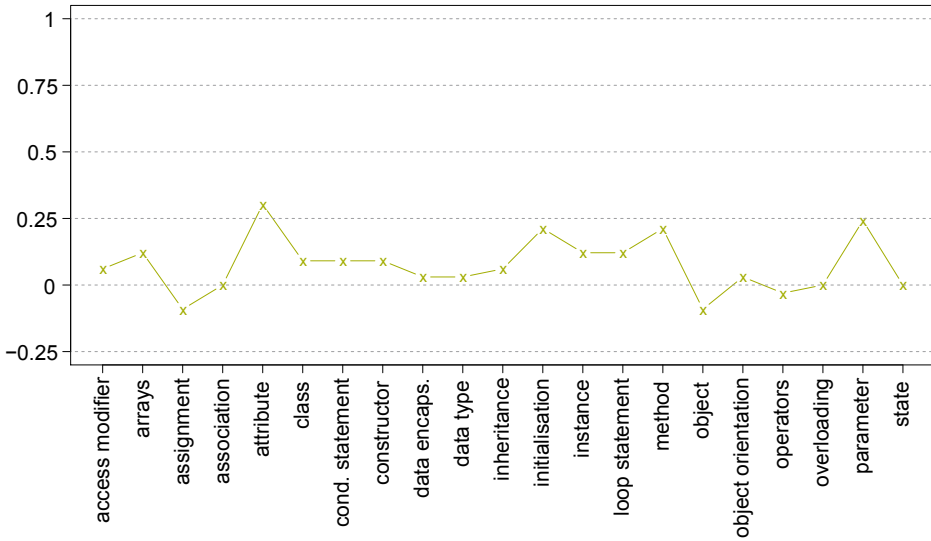


(a) No experience

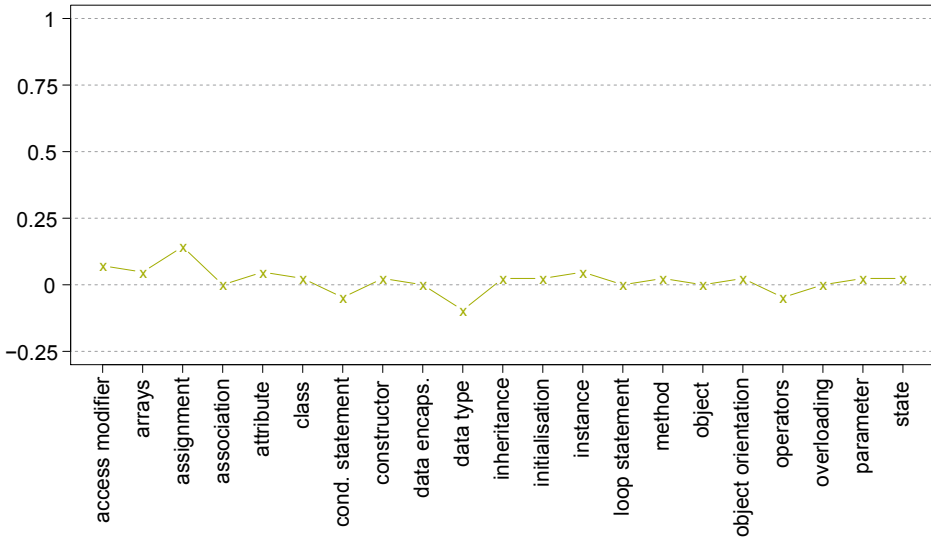


(b) Programming experience

Fig. 52: Development of structural knowledge dependent on prior programming experience. Shown is difference between post and pre test of the diagram in Fig. 51.



(a) No experience



(b) Programming experience

Fig. 53: Development of misconceptions dependent on prior programming experience. Shown is difference between post and pre test of the diagram in Fig. 42.



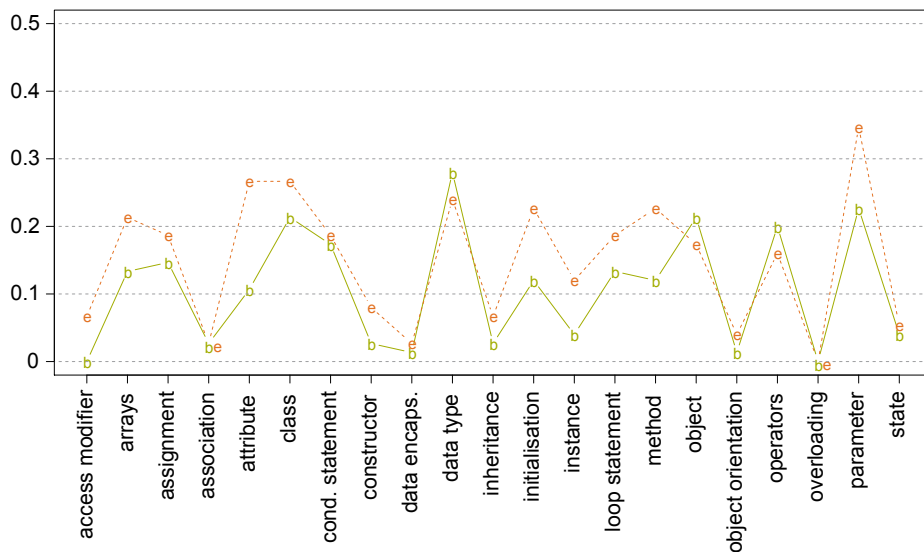


Fig. 54: Development of misconceptions over all participants. Shown is the percentage of maps that showed at least one “incorrect” edge (i.e. with a score of 2) incident to a concept (**b**(eginning) = pre test, **e**(nd) = post test).

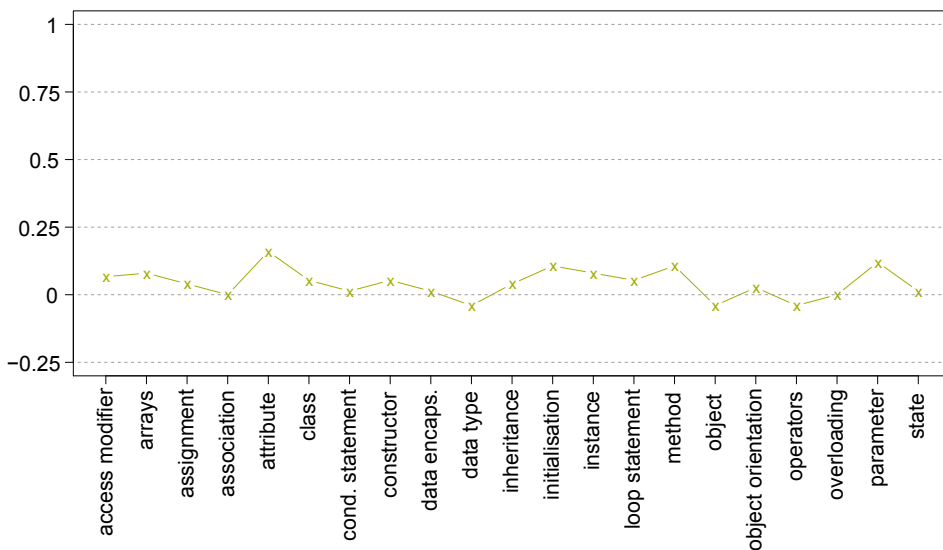
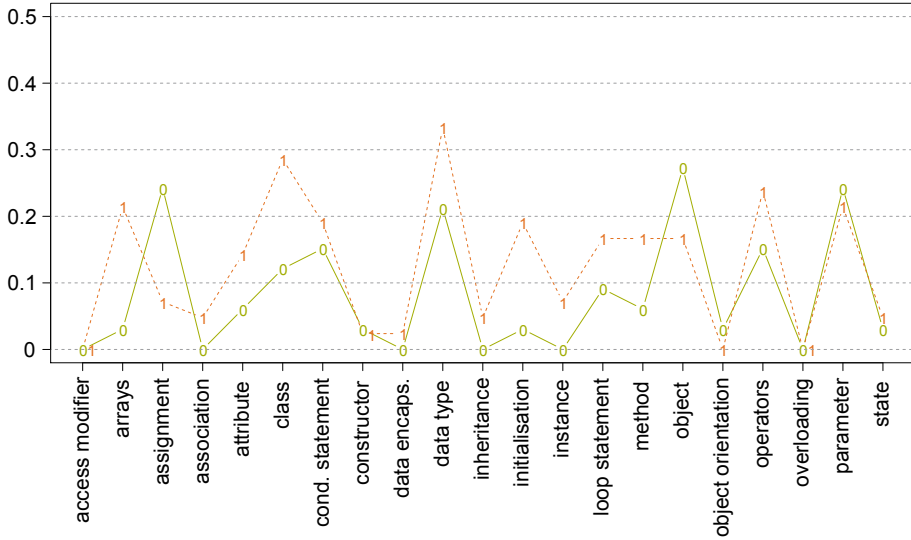
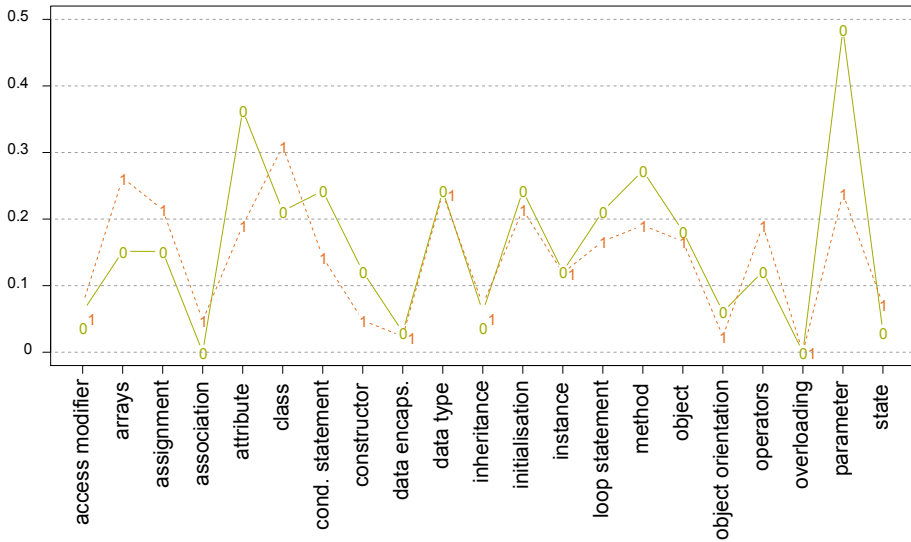


Fig. 55: Development of misconceptions over all participants. Shown is difference between post and pre test of the diagram in Fig. 54.

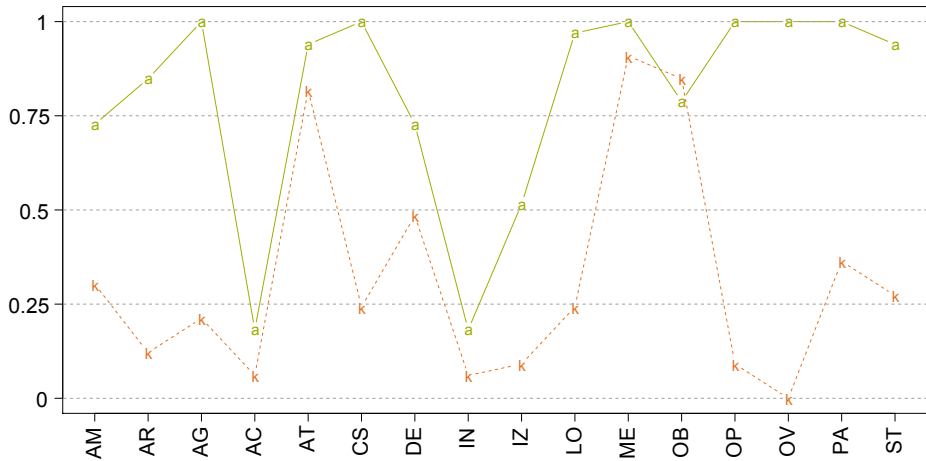


(a) Pre Maps

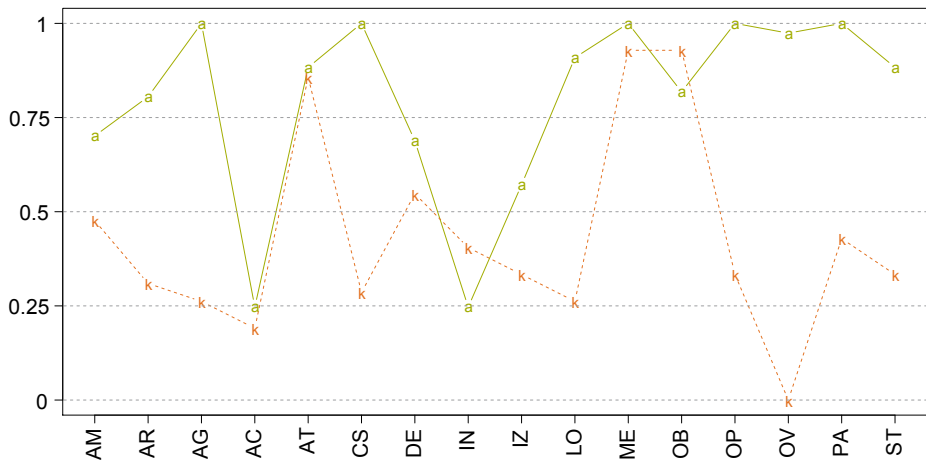


(b) Post Maps

Fig. 56: Development of misconceptions dependent on prior programming experience. Shown is the percentage of maps that showed at least one “incorrect” edge (i.e. with a score of 2) incident to a concept (0 = no prior experience, 1 = prior experience).



(a) Students without prior programming experience



(b) Students with prior programming experience

Fig. 57: Difference between conceptual knowledge and programming abilities concerning the concepts of CL (k = knowledge, a = abilities).

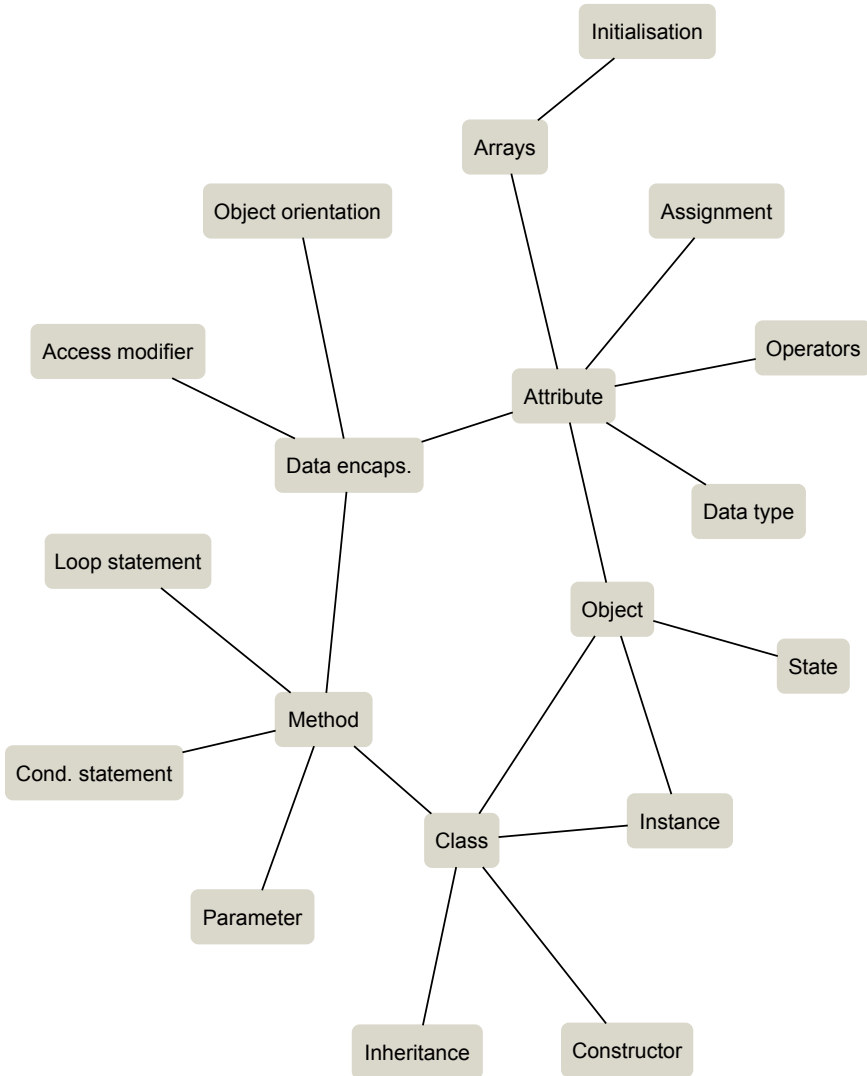


Fig. 58: The Pathfinder networks ( $q = n - 1$ ,  $r = \infty$ ) of the first cluster identified by using PAM clustering and graph similarity. Edges appearing in less than 10% of the maps have been pruned.

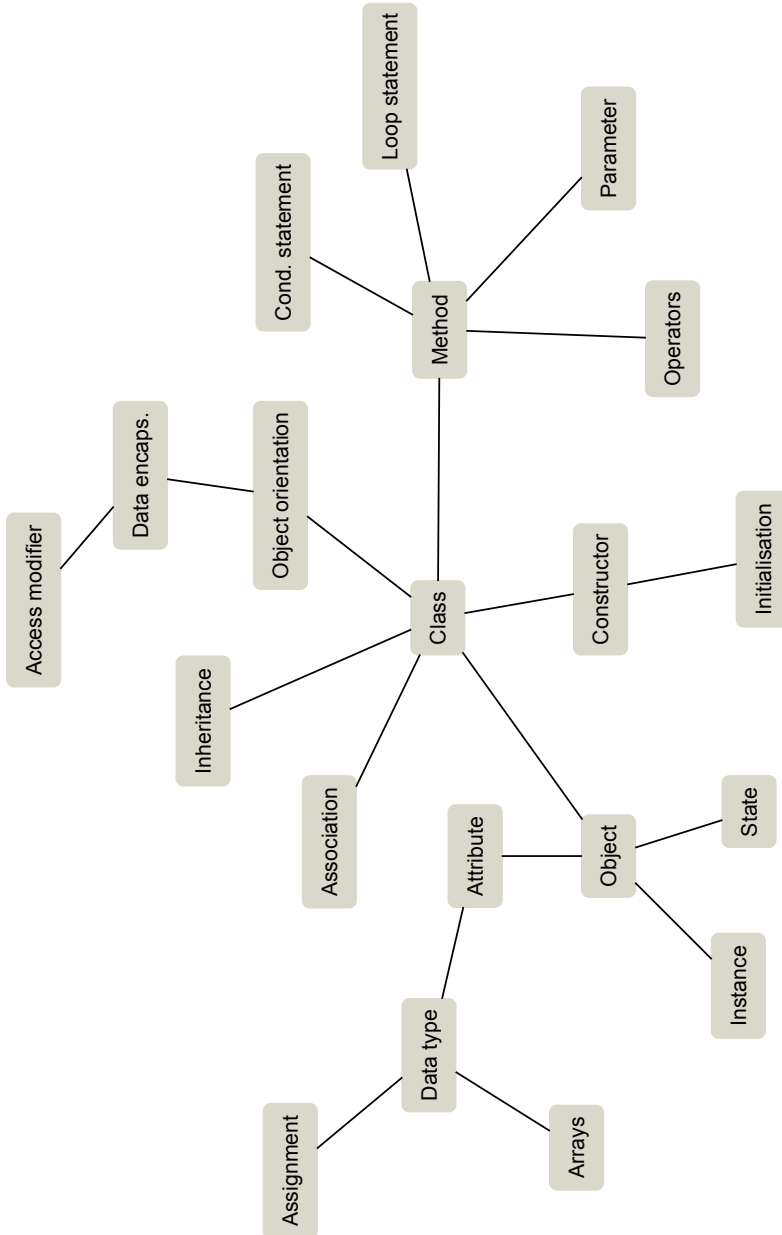


Fig. 59: The Pathfinder networks ( $q = n - 1, r = \infty$ ) of the second cluster identified by using PAM clustering and graph similarity. Edges appearing in less than 10% of the maps have been pruned.

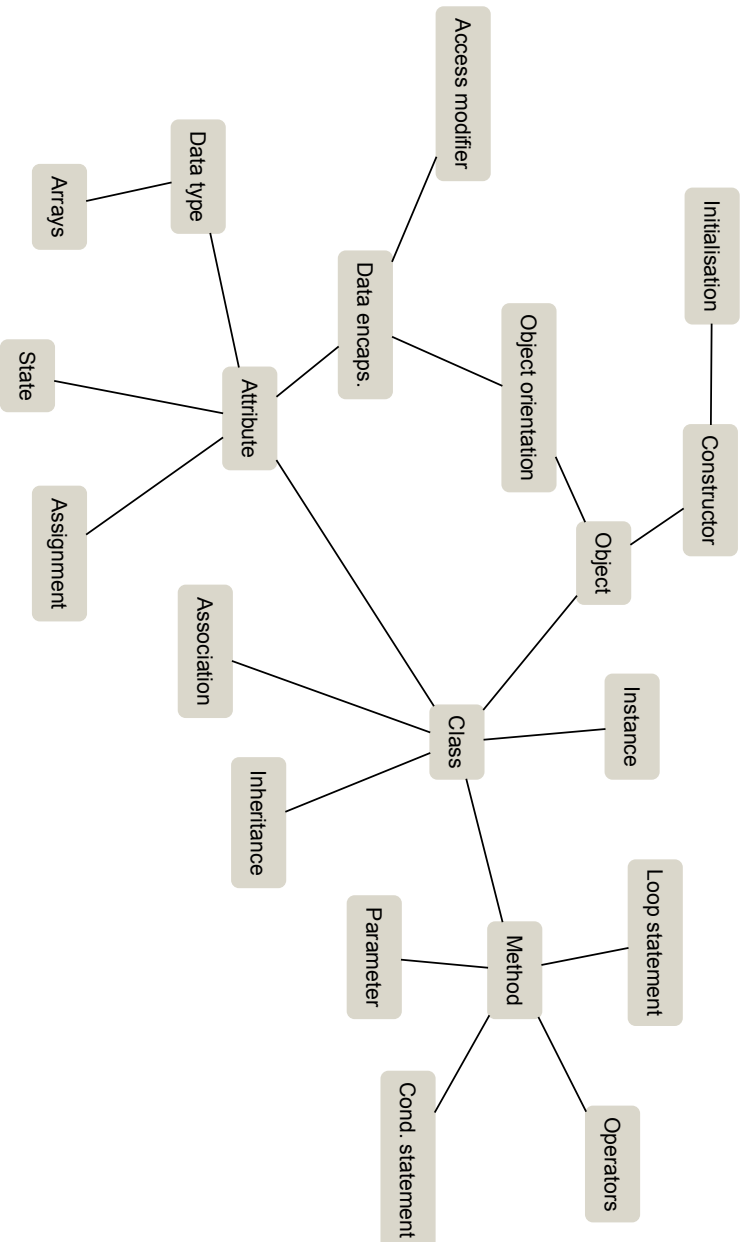


Fig. 60: The Pathfinder networks ( $q = n - 1$ ,  $r = \infty$ ) of the third cluster identified by using PAM clustering and graph similarity. Edges appearing in less than 10% of the maps have been pruned.

# List of Figures

Fig. 1	A general model for monitoring the effects of a learning stimulus on the knowledge of a person. Even though one is generally interested in the development in the left half of the diagram (i.e. the real person), only the right half is accessible to research. The transfers in form of an externalization and the analysis of the stimulus are in general not loss-less and are subject to many influences. . . . .	6
Fig. 2	The areas of research that provide the basis for this thesis. . . .	15
Fig. 3	Kolb's learning cycle . . . . .	37
Fig. 4	A model of learning, adapted from Hay et al. (2008, Fig. 7). All boxes with no outgoing arrows represent final states of the model: The two green ones on the left represent an outcome in which the student has learned something. The red box on the right represents an outcome of non-learning. . . . .	38
Fig. 5	A concept map illustrating the concept "proposition" and its relation to knowledge. Adapted from Novak (2010, p. 26) . . . . .	47
Fig. 6	The three different types of morphology of concept maps. Adapted from Kinchin (2000, p. 5-42) . . . . .	52
Fig. 7	Relational scoring method. Adapted from McClure et al. (1999, p. 482) . . . . .	64
Fig. 8	Schematic overview over the interconnections of the topics of chapter 3 and chapter 4 in the context of concept maps. . . . .	86
Fig. 9	The time between consecutive actions in the concept mapping editor for 58 (rather large) concept maps of students of two runs of the same lecture. The maps were drawn with no time restrictions and, in general, over several drawing sessions. . . . .	95
Fig. 10	A triadic proposition encoded in a concept map. . . . .	99

Fig. 11 A visualization of the two types of aggregations. Each 'x' marks the concept map of one person at one point in time. Some are missing. The aggregations either combine all maps of one point of measurement or all maps of one person. . . . . 114

Fig. 12 Accumulations still treat each map as an identifiable entity in the resulting data. Amalgamations result in a graph or concept map. In both cases, the newly formed data is the input for the subsequent analysis steps. . . . . 115

Fig. 13 A typical combination of aggregations: First, the maps of several points of measurement are amalgamated vertically, the resulting, new set of maps is then aggregated horizontally, using the times of measurements for the temporal ordering. . . . . 116

Fig. 14 Schematic overview over the different analysis methods and their relation to the different types of concept landscapes. . . . . 120

Fig. 15 MBMM and k-medoid clustering for the dataset of the first experiment in comparison . . . . . 127

Fig. 16 MBMM and k-medoid clustering for the dataset of the second experiment in comparison . . . . . 129

Fig. 17 The amalgamated concept landscape of 10 concept maps. The edge weights are transformed by using 11 minus the number of maps in which the given edge is present. The dashed edges are non-existent in the original concept maps and thus have an assumed weight of  $\infty$ . . . . . 135

Fig. 18 Pathfinder networks created from the input graph in Fig. 17 using four different sets of parameters. . . . . 137

Fig. 19 The result of applying the layout algorithm to a concept landscape taken from the first case study. It is based on communities that are aligned on a circle, the largest community forms the center. Even though the node placement has been optimized, there are several improvements that could be made. . . . . 147

Fig. 20 The three software projects that support the pivotal points of the schema presented in Fig. 1 . . . . . 149

Fig. 21 Screenshots of the front- and back end of CoMapEd. . . . . 154



Fig. 22	Screenshot of the graphical user interface of ConEx. . . . .	158
Fig. 23	Overview over the study, based on the schema of Fig. 1. . . . .	169
Fig. 24	An example map of MT2 as originally produced by one of the students in German. . . . .	171
Fig. 25	The first expert map containing all the propositions that were presented up to MT1. . . . .	174
Fig. 26	Development of basic graph measures. Plotted is the mean of each measure over the four points of measurement. The top and bottom lines show the standard deviation from the mean. Where these would have exceeded the minimal or maximal actually occurring value, those are used instead, indicating an asymmetrical distribution. . . . .	176
Fig. 27	The distribution of subgraphs of the four concepts <i>attribute</i> , <i>object</i> , <i>state</i> and <i>value</i> in the concept maps of MT1. . . . .	178
Fig. 28	The different subgraphs with more than one occurrence in MT1 - except for the empty subgraph. The numbers are in reference to Fig. 27 . . . . .	178
Fig. 29	Development of the high scoring propositions concerning OO related concepts. . . . .	180
Fig. 30	Development of the low scoring propositions (misconceptions) concerning OO related concepts. . . . .	181
Fig. 31	Overview over the study, based on the schema of Fig. 1. . . . .	190
Fig. 32	A digitalized example map originally produced by one of the students in German. . . . .	192
Fig. 33	The Pathfinder network of all the maps. . . . .	194
Fig. 34	The Pathfinder network of the first cluster identified by the PAM algorithm. . . . .	196
Fig. 35	The Pathfinder network of the second cluster identified by the PAM algorithm. . . . .	197
Fig. 36	The Pathfinder networks of those students who attended the compulsory school subject. . . . .	200

Fig. 37	The Pathfinder networks of students with no compulsory school subject. . . . .	201
Fig. 38	Overview over the study, based on the schema of Fig. 1. . . . .	207
Fig. 39	A scanned example map after that was produced by one of the students in German. . . . .	208
Fig. 40	Development of relevant structural knowledge over all participants. Shown is the fraction of maps with at least one “correct” edge (i.e. with a score of 2) incident to a concept ( <b>b</b> (eginning) = pre test, <b>e</b> (nd) = post test). . . . .	210
Fig. 41	Development of structural knowledge dependent on prior programming experience. Shown is the percentage of maps that showed at least one “correct” edge (i.e. with a score of 2) incident to a concept (0 = no prior experience, 1 = prior experience). . . . .	212
Fig. 42	Development of misconceptions dependent on prior programming experience. Shown is the percentage of maps that showed at least one “incorrect” edge ( <b>b</b> (eginning) = pre test, <b>e</b> (nd) = post test)). . . . .	213
Fig. 43	The probabilities of concept occurrence as identified by the MBMM clustering algorithm for the post maps, shown for both clusters. . . . .	214
Fig. 44	Pathfinder networks built from the post maps of the members of each cluster (as shown in Fig. 43). The parameter were set to $q = n - 1$ and $r = \infty$ . Before the creation, all concepts were removed that weren’t connected in more than half of the maps. . . . .	216
Fig. 45	Difference between conceptual knowledge and programming abilities concerning the concepts of CL (k = knowledge, a = abilities). . . . .	221
Fig. 46	The results of this thesis and how it affects the different research areas. . . . .	232
Fig. 47	The second expert map containing all the propositions that were presented between MT1 and MT2. . . . .	247
Fig. 48	The third expert map containing all the propositions that were presented between MT2 and the exam. . . . .	248

Fig. 49	The fourth expert map containing all the propositions that were presented between the exam and POST. . . . .	248
Fig. 50	Development of relevant structural knowledge over all participants. Shown is difference between post and pre test of the diagram in Fig. 40. . . . .	249
Fig. 51	Development of structural knowledge dependent on prior programming experience. Shown is the percentage of maps that showed at least one “correct” edge (i.e. with a score of 2) incident to a concept ( $\mathbf{b}(\text{eginning}) = \text{pre test}$ , $\mathbf{e}(\text{nd}) = \text{post test}$ ). . . . .	250
Fig. 52	Development of structural knowledge dependent on prior programming experience. Shown is difference between post and pre test of the diagram in Fig. 51. . . . .	251
Fig. 53	Development of misconceptions dependent on prior programming experience. Shown is difference between post and pre test of the diagram in Fig. 42. . . . .	252
Fig. 54	Development of misconceptions over all participants. Shown is the percentage of maps that showed at least one “incorrect” edge (i.e. with a score of 2) incident to a concept ( $\mathbf{b}(\text{eginning}) = \text{pre test}$ , $\mathbf{e}(\text{nd}) = \text{post test}$ ). . . . .	253
Fig. 55	Development of misconceptions over all participants. Shown is difference between post and pre test of the diagram in Fig. 54. . . . .	253
Fig. 56	Development of misconceptions dependent on prior programming experience. Shown is the percentage of maps that showed at least one “incorrect” edge (i.e. with a score of 2) incident to a concept (0 = no prior experience, 1 = prior experience). . . . .	254
Fig. 57	Difference between conceptual knowledge and programming abilities concerning the concepts of CL ( $k = \text{knowledge}$ , $a = \text{abilities}$ ). . . . .	255
Fig. 58	The Pathfinder networks ( $q = n - 1$ , $r = \infty$ ) of the first cluster identified by using PAM clustering and graph similarity. Edges appearing in less than 10% of the maps have been pruned. . . . .	256
Fig. 59	The Pathfinder networks ( $q = n - 1$ , $r = \infty$ ) of the second cluster identified by using PAM clustering and graph similarity. Edges appearing in less than 10% of the maps have been pruned. . . . .	257

Fig. 60	The Pathfinder networks ( $q = n - 1, r = \infty$ ) of the third cluster identified by using PAM clustering and graph similarity. Edges appearing in less than 10% of the maps have been pruned. . . .	258
---------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

## List of Tables

3.1	The levels of the SOLO taxonomy as described by Biggs & Collis (1982, Table 2.1) . . . . .	42
4.1	Differences between concepts maps of experts and novices as identified by Kinchin (2000, Table 5-4) . . . . .	53
5.1	The possible semantic relations in WordNet (cf. Miller 1995, Table 1). . . . .	81
6.1	Correlation between different elements of the assessment. Spearman's rank correlation was used (** = $p < 0.01$ , * = $p < 0.05$ , () = $p > 0.05$ ). . . . .	108
10.1	Spearman's rank correlation between the four basic graph measures. As score, the sum of all proposition scores of a map was used (** = $p < 0.01$ , () = $p > 0.05$ ). . . . .	175
11.1	The communities identified within the Pathfinder network from the maps of the G8 and G9 groups. The ordering of the communities for both groups is arbitrary, but was chosen to allow comparing the groups more easily. . . . .	199
12.1	The 21 concepts that the students should use for drawing the concept maps. The abbreviations are used later for the code analysis and in the diagrams of the results. . . . .	209

12.2	The dichotomous items used to analyze the code with regard to the application of the concepts of CL. The concepts are given as the abbreviations found in Table 12.1. . . . .	218
12.3	The items of CI and the percentile of projects that showed the respective property. . . . .	219

## References

- Åhlberg, Mauri (2004): Varieties of concept mapping, in A. J. Cañas, J. D. Novak, F. M. González, A. Cañas & F. M. González García (eds), *Concept maps: Theory, methodology, technology: Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, Sept 14-17 2004, Vol. 2, pp. 25–29.
- Akaike, H. (1974): A new look at the statistical model identification, *IEEE Transactions on Automatic Control* **19**(6): 716–723.
- Akinsanya, C. & Williams, M. (2004): Concept mapping for meaningful learning, *Nurse Education Today* **24**(1): 41–46.
- Al-Kunifed, Ali & Wandersee, James H. (1990): One hundred references related to concept mapping, *Journal of Research in Science Teaching* **27**(10): 1069–1075.
- Albert, D. & Steiner, C.M (2005): Empirical validation of concept maps: preliminary methodological considerations, in P. Goodyear, D. G. Sampson, D. J.-T. Yang, Kinshuk, T. Okamoto, R. Hartley & N.-S. Chen (eds), *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*, Kaohsiung, Taiwan, July 5 - 8, 2005, IEEE Computer Society Press, Los Alamitos, pp. 952–953.
- Altenmüller, E., Gruhn, W., Parlitz, D. & Liebert, G. (2000): The impact of music education on brain networks: evidence from eeg-studies, *International Journal of Music Education* **35**(1): 47–53.
- Anderson, Lorin W. & Krathwohl, David R. (2001): *A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational objectives*, complete ed., Longman, New York.

- Anohina-Naumeca, Alla (2012): Determining the set of concept map based tasks for computerized knowledge self-assessment, *Procedia - Social and Behavioral Sciences* **69**(0): 143–152.
- Anohina-Naumeca, Alla & Graudina, Vita (2012): Diversity of concept mapping tasks: Degree of difficulty, directedness, and task constraints, in A. J. Cañas, J. D. Novak & J. Vanhear (eds), *Concept Maps: Theory, Methodology, Technology: Proceedings of the Fifth International Conference on Concept Mapping*, Valletta, Malta, Sept 17-20 2012, Vol. 1, pp. 164–171.
- Anohina-Naumeca, Alla, Graudina, Vita & Grundspenkis, Janis (2012): Curricula comparison using concept maps and ontologies, in E. Stalidzans (ed.), *Proceedings of the 5th International Scientific Conference on Applied Information and Communication Technologies*, Jelgava, Latvia, April 26-27 2012, Faculty of Information Technologies, Latvia University of Agriculture, Jelgava, pp. 177–183.
- Anohina-Naumeca, Alla & Grundspenkis, Janis (2009): Scoring concept maps: an overview, in B. Rachev & A. Smrikarov (eds), *Proceedings of the International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*, Ruse, Bulgaria, June 18 - 19, 2009, Vol. 433 of *ACM International Conference Proceeding Series*, ACM, New York, pp. 78:1–78:6.
- Anohina-Naumeca, Alla, Grundspenkis, Janis & Strautmane, Maija (2011): The concept map-based assessment system: functional capabilities, evolution, and experimental results, *Int. J. Cont. Engineering Education and Life-Long Learning* **21**(4): 308–327.
- Armstrong, Deborah J. (2006): The quarks of object-oriented development, *Commun. ACM* **49**(2): 123–128.
- Ausubel, David Paul (1968): *Educational psychology: A cognitive view*, Holt, Rinehart et Winston, Montreal.
- Ausubel, David Paul (2000): *The acquisition and retention of knowledge: A cognitive view*, Kluwer Academic Publishers, Dordrecht and Boston.
- Baker, Ryan S. J. D. (2010): Data mining, in P. Peterson (ed.), *International encyclopedia of education*, Elsevier, Oxford, pp. 112–118.
- Baker, Ryan S. J. D. & Yacef, Kalina (2009): The state of educational data mining in 2009: A review and future visions, *JEDM - Journal of Educational Data Mining* **1**(1): 3–17.

- Balakrishnan, R. & Ranganathan, K. (2012): A Textbook of Graph Theory, Universitext, 2 ed., Springer, New York.
- Bartholomew, David J., Steele, Fiona, Moustaki, Irini & Galbraith, Jane I. (2008): Analysis of Multivariate Social Science Data, Chapman & Hall/CRC, Boca Raton.
- Berges, Marc, Mühling, Andreas & Hubwieser, Peter (2012): The gap between knowledge and ability, in M.-J. Laakso (ed.), Proceedings of the 12th Koli Calling International Conference on Computing Education Research, Koli, Finland, November 17-15 2012, ACM, New York, pp. 126–134.
- Berges, Marc, Mühling, Andreas, Hubwieser, Peter & Steuer, Horst (2013): Informatik für Nichtinformatiker: ein kontext- und praxisorientiertes Konzept, in P. Forbrig, D. Rick & A. Schmolitzky (eds), HDI 2012 – Informatik für eine nachhaltige Zukunft, 5. Fachtagung Hochschuldidaktik der Informatik, Hamburg, Germany, November 6-7 2012, Vol. 5 of *Commentarii informaticae didacticae*, Universitätsverlag Potsdam, Potsdam, pp. 105–110.
- Bergin, Joe, Bruce, Kim & Kölling, Michael (2005): Objects-early tools: A demonstration, *SIGCSE Bull* **37**(1): 390–391.
- Biggs, John Burville & Collis, Kevin F. (1982): Evaluating the quality of learning: The SOLO taxonomy (structure of the observed learning outcome), Educational psychology, Academic Press, New York.
- Biggs, John Burville & Tang, Catherine So-kum (2011): Teaching for quality learning at university, 4 ed., Open University Press, Maidenhead.
- Black, P. & Wiliam, D. (1998): Assessment and classroom learning, *Assessment in Education: Principles, Policy & Practice* **5**(1): 7–74.
- Bloom, Benjamin Samuel (1956): Taxonomy of educational objectives: The classification of educational goals, David McKay, New York.
- Börstler, Jürgen, Christensen, Henrik B., Bennedsen, Jens, Nordström, Marie, Kallin Westin, Lena, Moström, Jan Erik & Caspersen, Michael E. (2008): Evaluating oo example programs for cs1, in J. Amillo, C. Laxer, E. M. Ruiz & A. Young (eds), Proceedings of the 13th annual conference on Innovation and technology in computer science education, Madrid, Spain, June 30-July 2, 2008, ITiCSE '08, ACM, New York, pp. 47–52.
- Bortz, Jürgen (2005): Statistik: Für Human- und Sozialwissenschaftler, Springer, Heidelberg.

- Breier, Norbert & Hubwieser, Peter (2002): An information-oriented approach to informatical education, *Informatics in education* 1(1): 31–42.
- Cañas, Alberto J. (2008): A semantic scoring rubric for concept maps: Design and reliability, in A. J. Cañas, P. Reiska, M. Åhlberg & J. D. Novak (eds), *Concept Mapping: Connecting Educators: Proceedings of the Third International Conference on Concept Mapping*, Tallinn, Estonia & Helsinki, Finland, September 22-25 2008, Vol. 1, Tallinn University, Estonia, pp. 60–67.
- Cañas, Alberto J., Bunch, Larry & Reiska, P. (2010): Cmapanalysis: An extensible concept map analysis tool, in J. Sánchez, A. J. Cañas & J. D. Novak (eds), *Concept Maps: Making Learning Meaningful: Proceedings of the Fourth International Conference on Concept Mapping*, Viña del Mar, Chile, October 5-7 2010, Vol. 1, Universidad de Chile, Chile, pp. 75–83.
- Cañas, Alberto J., Carff, Roger, Hill, Greg, Carvalho, Marco, Arguedas, Marco, Eskridge, Tom, Lott, James & Carvajal, Rodrigo (2005): Concept maps: Integrating knowledge and information visualization, in S.-O. Tergan & T. Keller (eds), *Knowledge and Information Visualization*, Vol. 3426 of *Lecture notes in computer science*, Springer Berlin Heidelberg, pp. 205–219.
- Cañas, Alberto J., Hill, Greg, Carff, Roger, Suri, Niranjana, Lott, James & Gómez, Gloria (2004): Cmaptools: A knowledge modeling and sharing environment, in A. J. Cañas, J. D. Novak, F. M. González, A. Cañas & F. M. González García (eds), *Concept maps: Theory, methodology, technology: Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, Sept 14-17 2004, Vol. 1, pp. 125–134.
- Cañas, Alberto J. & Novak, Joseph D. (2006): Re-examining the foundations for effective use of concept maps, in A. J. Cañas & J. D. Novak (eds), *Concept Maps: Theory, Methodology, Technology: Proceedings of the Second International Conference on Concept Mapping*, San José, Costa Rica, September 5-6 2006, Vol. 1, Universidad de Costa Rica, San José and Costa Rica, pp. 494–502.
- Cañas, Alberto J. & Novak, Joseph D. (2012): Freedom vs. restriction of content and structure during concept mapping - possibilities and limitations for construction and assessment, in A. J. Cañas, J. D. Novak & J. Vanhear (eds), *Concept Maps: Theory, Methodology, Technology: Proceedings of the Fifth International Conference on Concept Mapping*, Valletta, Malta, Sept 17-20 2012, Vol. 2, pp. 247–257.
- Cañas, Alberto J., Valerio, Alejandro, Lalinde-Pulido, Juan, Carvalho, Marco & Arguedas, Marco (2003): Using wordnet for word sense disambiguation to



- support concept map construction, in M. A. Nascimento, E. S. Moura & A. L. Oliveira (eds), *String Processing and Information Retrieval*, Vol. 2857 of *Lecture notes in computer science*, Springer Berlin Heidelberg, pp. 350–359.
- Chen, Chaomei (2002): Information visualization, *Information Visualization* **1**(1): 1–4.
- Chiu, Chiung-Hui, Huang, Chun-Chieh & Chang, Wen-Tsung (2000): The evaluation and influence of interaction in network supported collaborative concept mapping, *Computers & Education* **34**(1): 17–25.
- Clariana, Roy B. & Taricani, Ellen M. (2010): The consequences of increasing the number of terms used to score open ended concept maps, *International Journal of Instructional Media* **37**(2): 163–172.
- Clauset, Aaron, Moore, Christopher & Newman, M. E. J. (2007): Structural inference of hierarchies in networks, in E. Airoldi (ed.), *Statistical network analysis: models, issues, and new directions*, Vol. 4503 of *Lecture notes in computer science*, Springer, Berlin, pp. 1–13.
- Clauset, Aaron, Newman, M. E. J. & Moore, Christopher (2004): Finding community structure in very large networks, *Physical Review E* **70**(6): 066111.
- Cooke, Nancy Jaworski (1983): Memory structures of expert and novice computer programmers: Recall order vs. similarity ratings, PhD thesis, New Mexico State University, New Mexico.
- Cooke, Nancy Jaworski (1990): Using pathfinder as a knowledge elicitation tool: Link interpretation, in R. W. Schvaneveldt (ed.), *Pathfinder associative networks*, Ablex Pub. Corp., Norwood, pp. 227–239.
- Cooke, Nancy Jaworski (1994): Varieties of knowledge elicitation techniques, *International Journal of Human-Computer Studies* **41**(6): 801–849.
- Cormen, Thomas H., Leiserson, Charles Eric, Rivest, Ronald L. & Stein, Clifford (2001): *Introduction to algorithms*, 2nd ed., MIT Press and McGraw-Hill, Cambridge and New York.
- Daley, B. J., Conceicao, S. C. O., Mina, L., Altman, B. A., Baldor, M. & Brown, J. (2010): Integrative literature review: Concept mapping: A strategy to support the development of practice, research, and theory within human resource development, *Human Resource Development Review* **9**(4): 357–384.

- Dayton, Tom, Durso, Francis T. & Shepard, Jack D. (1990): A measure of the knowledge reorganization under insight, in R. W. Schvaneveldt (ed.), *Pathfinder associative networks*, Ablex Pub. Corp., Norwood, pp. 267–277.
- de Jong, Ton & Ferguson-Hessler, Monica G.M. (1996): Types and qualities of knowledge., *Educational Psychologist* **31**(2): 105–113.
- Dearholt, Donald W. & Schvaneveldt, Roger W. (1990): Properties of pathfinder networks, in R. W. Schvaneveldt (ed.), *Pathfinder associative networks*, Ablex Pub. Corp., Norwood, pp. 1–30.
- Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977): Maximum likelihood from incomplete data via the em maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1): 1–38.
- Derbentseva, Natalia, Safayeni, Frank & Cañas, Alberto J. (2007): Concept maps: Experiments on dynamic thinking, *Journal of Research in Science Teaching* **44**(3): 448–465.
- Dierkes, Meinolf (2001): *Handbook of organizational learning and knowledge*, Oxford University Press, Oxford.
- Diethelm, Ira, Hubwieser, Peter & Klaus, Robert (2012): Students, teachers and phenomena: Educational reconstruction for computer science education, in M.-J. Laakso (ed.), *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*, Koli, Finland, November 17-15 2012, ACM, New York, pp. 164–173.
- Doignon, Jean-Paul & Falmagne, Jean-Claude (1985): Spaces for the assessment of knowledge, *International Journal of Man-Machine Studies* **23**: 175–196.
- Duit, Reinders & Treagust, David F. (2003): Conceptual change: A powerful framework for improving science teaching and learning, *International Journal of Science Education* **25**(6): 671–688.
- Durso, Francis T. & Coggins, Kathy A. (1990): Graphs in the social and psychological sciences: Empirical contributions of pathfinder, in R. W. Schvaneveldt (ed.), *Pathfinder associative networks*, Ablex Pub. Corp., Norwood, pp. 31–51.
- Eckerdal, Anna (2006): *Novice Students' Learning of Object-Oriented Programming*, PhD thesis, Uppsala Universitet, Uppsala.

- Eckerdal, Anna, McCartney, Robert, Moström, Jan Erik, Ratcliffe, Mark, Sanders, Kate & Zander, Carol (2006): Putting threshold concepts into context in computer science education, *SIGCSE Bull* **38**(3): 103–107.
- Eckert, Andreas (2000): Die Netzwerk-Elaborierungs-Technik (NET) - Ein computerunterstütztes Verfahren zur Diagnose komplexer Wissensstrukturen, in H. Mandl & F. Fischer (eds), *Wissen sichtbar machen*, Hogrefe, Göttingen, pp. 137–157.
- Edmondson, Katherine M. (2005): Assessing science understanding through concept maps, in J. J. Mintzes (ed.), *Assessing science understanding*, Elsevier, Amsterdam, pp. 15–40.
- Ertl, Bernhard & Mok, Sog Yee (2010): Concept-mapping im informatikunterricht, *Log in* (166/167): 63–68.
- Eynard, Davide, Marfia, Fabio & Matteucci, Matteo (2010): On the use of correspondence analysis to learn seed ontologies from text, in J. Filipe & J. L. G. Dietz (eds), *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, Valencia, Spain, October 25-28, 2010, SciTePress, pp. 430–439.
- Fellbaum, Christiane (1999): *WordNet: An electronic lexical database*, Language, speech, and communication, 2nd ed., MIT Press, Cambridge.
- Fincher, Sally & Petre, Marian (2004): *Computer science education research*, RoutledgeFalmer, London and New York.
- Forte, Andrea & Guzdial, Mark (2004): Computers for communication, not calculation: Media as a motivation and context for learning, in R. H. Sprague (ed.), *Proceedings of the 37th Hawaii International Conference on System Sciences (CD-ROM)*, 4 - 7 Januar 2012, IEEE, Piscataway.
- Freeman, Linton C. (1978): Centrality in social networks conceptual clarification, *Social Networks* **1**(1): 215–239.
- Fuller, Ursula, Johnson, Colin G., Ahoniemi, Tuukka, Cukierman, Diana, Hernán-Losada, Isidoro, Jackova, Jana, Lahtinen, Essi, Lewis, Tracy L., Thompson, Donna McGee, Riedesel, Charles & Thompson, Errol (2007): Developing a computer science-specific learning taxonomy, *Working group reports on Innovation and technology in computer science education, ITiCSE-WGR '07*, ACM, New York, pp. 152–170.

- Gammack, John G. (1990): Expert conceptual structure: The stability of pathfinder representations, in R. W. Schvaneveldt (ed.), *Pathfinder associative networks*, Ablex Pub. Corp., Norwood, pp. 212–226.
- Geber, Beryl A. (2006): *Piaget and knowing: Studies in genetic epistemology*, Routledge, London.
- Glaser, Robert & Bassok, Miriam (1989): Learning theory and the study of instruction, *Annual Review of Psychology* **40**: 631–666.
- Glaserfeld, Ernst von (1983): Learning as a constructive activity, in J. C. Bergeron (ed.), *Proceedings of the 5th Annual Meeting of the North American Group of Psychology in Mathematics Education*, Montreal, pp. 41–101.
- Glaserfeld, Ernst von (1989): Constructivism in education, in T. Husén & T. N. Postlethwaite (eds), *The international encyclopedia of education*, Pergamon Press, Oxford, pp. 162–163.
- Glöggler, Karl (1997): Handlungsorientierter Unterricht im Berufsfeld Elektrotechnik: Untersuchung einer Konzeption in der Berufsschule und Ermittlung der Veränderung expliziten Handlungswissens, Vol. 16 of *Beiträge zur Arbeits-, Berufs- und Wirtschaftspädagogik*, P. Lang, Frankfurt am Main and New York.
- Goldsmith, Timothy E. & Davenport, Daniel M. (1990): Assessing structural similarity of graphs, in R. W. Schvaneveldt (ed.), *Pathfinder associative networks*, Ablex Pub. Corp., Norwood, pp. 74–87.
- Goldsmith, Timothy E. & Johnson, Peder J. (1990): A structural assessment of classroom learning, in R. W. Schvaneveldt (ed.), *Pathfinder associative networks*, Ablex Pub. Corp., Norwood, pp. 240–254.
- Goldstein, Eugen Bruce & Vanhorn, Daniel (2011): *Cognitive psychology*, 3 ed., Wadsworth Cengage Learning, Australia and Belmont.
- Goldstone, Robert L. & Kersten, Alan (2003): Concepts and categorization, in I. B. Weiner (ed.), *Handbook of psychology*, Vol. 4, John Wiley, Hoboken, pp. 599–621.
- Gordon, A. D. (1999): Classification, Vol. 82 of *Monographs on statistics and applied probability*, 2 ed., Chapman & Hall/CRC, Boca Raton.
- Gouli, Evangelia (2007): *Concept Mapping in Didactics of Informatics. Assessment as a Tool for Learning in Web-based and Adaptive Educational Environments*, PhD thesis, National and Kapodistrian University of Athens, Athen.

- Gouli, Evangelia, Gogoulou, Agoritsa, Papanikolaou, Kyparisia A. & Grigoriadou, Maria (2005a): How to qualitatively + quantitatively assess concepts maps: the case of compass, in C.-K. Looi (ed.), *Artificial intelligence in education*, IOS Press, Amsterdam, pp. 804–806.
- Gouli, Evangelia, Gogoulou, Agoritsa, Papanikolaou, Kyparisia & Grigoriadou, Maria (2005b): Evaluating learner’s knowledge level on concept mapping tasks, in P. Goodyear, D. G. Sampson, D. J.-T. Yang, Kinshuk, T. Okamoto, R. Hartley & N.-S. Chen (eds), *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*, Kaohsiung, Taiwan, July 5 - 8, 2005, IEEE Computer Society Press, Los Alamitos and California, pp. 424–428.
- Gries, David (2008): A principled approach to teaching oo first, *SIGCSE Bull* **40**(1): 31–35.
- Grundspenkis, Janis & Strautmane, Maija (2010): Usage of graph patterns for knowledge assessment based on concept maps, *Scientific Journal of Riga Technical University. Computer Sciences* **38**(39): 60–71.
- Guerrero-Bote, Vicente P., Zapico-Alonso, Felipe, Espinosa-Calvo, María Eugenia, Gómez Crisóstomo, Rocío & Moya-Anegón, Félix de (2006): Binary pathfinder: An improvement to the pathfinder algorithm, *Information Processing & Management* **42**(6): 1484–1490.
- Gureckis, Todd M. & Goldstone, Robert L. (2011): Schema, in P. C. Hogan (ed.), *The Cambridge encyclopedia of the language sciences*, Cambridge Univ. Press, Cambridge, pp. 725–726.
- Gurlitt, Johannes & Renkl, Alexander (2010): Prior knowledge activation: how different concept mapping tasks lead to substantial differences in cognitive processes, learning outcomes, and perceived self-efficacy, *Instructional Science* **38**(4): 417–433.
- Guzdial, Mark (2003): A media computation course for non-majors, in V. Dagdilelis, M. Satratzemi, R. Boyle, G. Evangelidis & D. Finkel (eds), *Proceedings of the 8th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, Thessaloniki, Greece, June 30-July 2, 2003, Association for Computing Machinery, New York, pp. 104–108.
- Han, Jiawei & Kamber, Micheline (2010): *Data mining: Concepts and techniques*, The Morgan Kaufmann series in data management systems, 2 ed., Elsevier/Morgan Kaufmann, Amsterdam.

- Hay, David B. & Kinchin, Ian M. (2006): Using concept maps to reveal conceptual typologies, *Education + Training* **48**(2/3): 127–142.
- Hay, David B., Kinchin, Ian M. & Lygo-Baker, Simon (2008): Making learning visible: the role of concept mapping in higher education, *Studies in Higher Education* **33**(3): 295–311.
- Hoffman, Robert R., Shadbolt, Nigel R., Burton, A. Mike & Klein, Gary (1995): Eliciting knowledge from experts: A methodological analysis, *Organizational Behavior and Human Decision Processes* **62**(2): 129–158.
- Holme, P. (2003): Network dynamics of ongoing social relationships, *EPL (Europhysics Letters)* **64**: 427–433.
- Hubwieser, Peter (2007): A smooth way towards object oriented programming in secondary schools, in I. M. e. . Benzie D. (ed.), *Informatics, Mathematics and ICT: a 'golden triangle'*, College of Computer and Information Science Northeastern University Boston, Boston.
- Hubwieser, Peter (2008): Analysis of learning objectives in object oriented programming, in R. T. Mittermeir & M. M. Sysło (eds), *Informatics Education - Supporting Computational Thinking*, Vol. 5090 of *Lecture notes in computer science*, Springer Berlin Heidelberg, pp. 142–150.
- Hubwieser, Peter (2012): Computer science education in secondary schools – the introduction of a new compulsory subject, *Trans. Comput. Educ.* **12**(4): 16:1–16:41.
- Hubwieser, Peter & Berges, Marc (2011): Minimally invasive programming courses: learning oop with(out) instruction, in T. J. Cortina, E. L. Walker, L. S. King, D. R. Musicant & L. I. McCann (eds), *Proceedings of the 42nd ACM technical symposium on Computer science education*, Dallas, USA, March 9-12 2011, ACM, New York, pp. 87–92.
- Hubwieser, Peter, Magenheim, Johannes, Mühling, Andreas & Ruf, Alexander (2013): Towards a conceptualization of pedagogical content knowledge for computer science, in B. Simon, A. Clear & Q. I. Cutts (eds), *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, La Jolla, CA, USA, August 12-14, 2013, ACM, New York, pp. 1–8.
- Hubwieser, Peter & Mühling, Andreas (2011a): Investigating cognitive structures of object oriented programming, *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, Darmstadt, Germany, June 27-29 2011, ACM, New York, p. 377.

- Hubwieser, Peter & Mühling, Andreas (2011b): Knowpats: Patterns of declarative knowledge - searching frequent knowledge patterns about object-orientation, in J. Filipe & Ana L. N. Fred (eds), KDIR 2011 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, Paris, France, 26-29 October, 2011, SciTePress, pp. 358–364.
- Hubwieser, Peter & Mühling, Andreas (2011c): What students (should) know about object oriented programming, in K. Sanders, M. E. Caspersen & A. Clear (eds), Proceedings of the Seventh International Workshop on Computing Education Research, Rhode Island, USA, August 8-9, 2011, ACM, New York, pp. 77–84.
- Ifenthaler, Dirk (2006): Diagnose lernabhängiger Veränderung mentaler Modelle, PhD thesis, Albert-Ludwigs-Universität, Freiburg.
- Îngeç, Şebnem Kandil (2009): Analysing concept maps as an assessment tool in teaching physics and comparison with the achievement tests, *International Journal of Science Education* **31**(14): 1897–1915.
- Jarvis, Peter (1992): Paradoxes of learning: On becoming an individual in society, The Jossey-Bass social and behavioral science series, 1 ed., Jossey-Bass, San Francisco.
- Jarvis, Peter (2012): Adult learning in the social context, Vol. 78 of *Routledge library editions, Education*, Routledge, London and New York.
- Karagiorgi, Yiasemina & Symeou, Loizos (2005): Translating constructivism into instructional design: Potential and limitations, *Educational Technology & Society* **8**(1): 17–27.
- Karakuyu, Yunus (2010): The effect of concept mapping on attitude and achievement in a physics course, *International Journal of the Physical Sciences* **5**(6): 724–737.
- Kaufman, Leonard & Rousseeuw, Peter J. (2005): Finding groups in data: An introduction to cluster analysis, Wiley-Interscience paperback series, Wiley, Hoboken.
- Kaufmann, Michael & Wagner, Dorothea (2001): Drawing graphs: Methods and models, Vol. 2025 of *Lecture notes in computer science*, Springer, Berlin and New York.
- Keppens, Jeroen & Hay, David (2008): Concept map assessment for teaching computer programming, *Computer Science Education* **18**(1): 31–42.

- Kern, Cindy & Crippen, Kent J (2008): Mapping for conceptual change, *Science Teacher* **75**(6): 32–38.
- Kinchin, Ian & Hay, David (2007): The myth of the research-led teacher, *Teachers and Teaching* **13**(1): 43–61.
- Kinchin, Ian M. (2000): The active use of concept mapping to promote meaningful learning in biological science, PhD thesis, University of Surrey, Guilford.
- Kinchin, Ian M. (2011): Visualising knowledge structures in biology: discipline, curriculum and student understanding, *Journal of Biological Education* **45**(4): 183–189.
- Kinchin, Ian M. (2013): Concept mapping and the fundamental problem of moving between knowledge structures, *Journal for Educators, Teachers and Trainers* **4**(1): 96–106.
- Kinchin, Ian M. & Cabot, L. B. (2009): An introduction to concept mapping in dental education: the case of partial denture design, *European Journal of Dental Education* **13**(1): 20–27.
- Kinchin, Ian M., Cabot, L. B., Kobus, M. & Woolford, M. (2011): Threshold concepts in dental education, *European Journal of Dental Education* **15**(4): 210–215.
- Kinchin, Ian M., Hay, David B. & Adams, Alan (2000): How a qualitative approach to concept map analysis can be used to aid learning by illustrating patterns of conceptual development, *Educational Research* **42**(1): 43–57.
- Kirkpatrick, S., Gelatt, Jr. C. D. & Vecchi, M. P. (1983): Optimization by simulated annealing, *Science* **220**(4598): 671–680.
- Klieme, Eckhard, Hartig, Johannes & Rauch, Dominique (2008): The concept of competence in educational contexts, in E. Klieme, D. Leutner & J. Hartig (eds), *Assessment of competencies in educational contexts*, Hogrefe & Huber Publishers, Toronto.
- Knobelsdorf, Maria (2011): Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung, PhD thesis, Freie Universität Berlin, Berlin.
- Kolb, D. A. (1984): *Experiential learning: experience as the source of learning and development*, Prentice Hall, Englewood Cliffs.
- Kolb, David Allen & Fry, Ronald E. (1975): Toward an applied theory of experiential learning, in C. L. Cooper (ed.), *Theories of group processes*, Wiley series on individuals, groups and organizations, Wiley, London, pp. 33–110.



- Koponen, Ismo T. & Pehkonen, Maija (2010): Entropy and energy in characterizing the organization of concept maps in learning science, *Entropy* **12**(7): 1653–1672.
- Kornilakis, Harry, Grigoriadou, Maria, Papanikolaou, Kyparisia A. & Gouli, Evangelia (2004): Using wordnet to support interactive concept map construction, in Kinshuk, C.-K. Looi, E. Sutinen, D. G. Sampson, I. Aedo, L. Uden & E. Kähkönen (eds), *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, Joensuu, Finland, 30 August - 1 September 2004., IEEE Computer Society, Los Alamitos, pp. 600–604.
- Koul, Ravinder, Clariana, Roy B. & Salehi, Roya (2005): Comparing several human and computer-based methods for scoring concept maps and essays, *Journal of Educational Computing Research* **32**(3): 227–239.
- Kwon, So Young & Cifuentes, Lauren (2009): The comparative effect of individually-constructed vs. collaboratively-constructed computer-based concept maps, *Computers & Education* **52**(2): 365–375.
- Larraza-Mendiluze, Edurne & Garay-Vitoria, Nestor (2013): Use of concept maps to analyze students' understanding of the i/o subsystem, in M.-J. Laakso & Simon (eds), *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, Koli, Finland, November 14-17 2013, Koli Calling '13, ACM, New York, pp. 67–76.
- Lawler, Eugene L. (2001): *Combinatorial optimization: Networks and matroids*, Dover Publications, Mineola.
- Lawson, Richard G. & Jurs, Peter C. (1990): New index for clustering tendency and its application to chemical problems, *Journal of Chemical Information and Computer Sciences* **30**(1): 36–41.
- Leake, David B., Maguitman, Ana & Cañas, Alberto J. (2002): Assessing conceptual similarity to support concept mapping, in S. M. Haller & G. Simmons (eds), *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, Pensacola Beach, USA, May 14-16 2002, AAAI Press, pp. 168–172.
- Leake, David B., Maguitman, Ana & Reichherzer, Thomas (2005): Understanding knowledge models: Modeling assessment of concept importance in concept maps, in K. Forbus, D. Gentner & T. Regier (eds), *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society*, Chicago, USA, August 4-7 2004, Lawrence Erlbaum Associates, Mahwah, pp. 785–800.

- Leake, David B., Reichherzer, Thomas, Cañas, Alberto J., Carvalho, Marco & Eskridge, Tom (2004): "googling" from a concept map: Towards automatic concept-map-based query formation, in A. J. Cañas, J. D. Novak, F. M. González, A. Cañas & F. M. González García (eds), *Concept maps: Theory, methodology, technology: Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, Sept 14-17 2004, Vol. 1, pp. 409–416.
- Linck, B., Ohrndorf, L., Schubert, S., Stechert, P., Magenheimer, J., Nelles, W., Neugebauer, J. & Schaper, N. (2013): Competence model for informatics modelling and system comprehension, 2013 IEEE Global Engineering Education Conference, Berlin, Germany, March 13-15 2013, IEEE, Piscataway, pp. 85–93.
- Ma, X. (2010): Longitudinal evaluation designs, in P. Peterson (ed.), *International encyclopedia of education*, Elsevier, Oxford, pp. 757–764.
- Malmi, Lauri, Sheard, Judy, Simon, Bednarik, Roman, Helminen, Juha, Korhonen, Ari, Myller, Niko, Sorva, Juha & Taherkhani, Ahmad (2010): Characterizing research in computing education: a preliminary analysis of the literature, *Proceedings of the Sixth international workshop on Computing education research*, Aarhus, Denmark, August 9-10 2010, ACM, New York, pp. 3–12.
- Mandl, Heinz & Fischer, Frank (2000): Mapping-Techniken und Begriffsnetze in Lern- und Kooperationsprozessen, in H. Mandl & F. Fischer (eds), *Wissen sichtbar machen*, Hogrefe, Göttingen, pp. 3–12.
- Mazur, Eric (1996): Are science lectures a relic of the past?, *Physics World* **9**: 13–14.
- McClelland, David C. (1973): Testing for competence rather than for "intelligence", *American Psychologist* **28**(1): 1–14.
- McClure, John R., Sonak, Brian & Suen, Hoi K. (1999): Concept map assessment of classroom learning: Reliability, validity, and logistical practicality, *Journal of Research in Science Teaching* **36**(4): 475–492.
- Mehta, Dinesh P. (2005): *Handbook of data structures and applications*, computer and information science series, Chapman & Hall/CRC, Boca Raton.
- Meyer, Jan & Land, Ray (2006): *Overcoming barriers to student understanding: Threshold concepts and troublesome knowledge*, Routledge, London and New York.
- Miller, George A. (1995): Wordnet: a lexical database for english, *Communications of the ACM* **38**(11): 39–41.

- Miller, Norma L. & Cañas, Alberto J. (2008): Effect of the nature of the focus question on presence of dynamic propositions in a concept map, in A. J. Cañas, P. Reiska, M. Åhlberg & J. D. Novak (eds), *Concept Mapping: Connecting Educators: Proceedings of the Third International Conference on Concept Mapping: Tallinn, Estonia & Helsinki, Finland: September 22-25 2008*, Vol. 1, Tallinn University, Estonia, pp. 365–372.
- Mistades, Voltaire Mallari (1999): Concept mapping in introductory physics, *Journal of Education and Human Development* **3**(1).
- Mitchell, Marcus P., Santorini, Beatrice & Marcinkiewicz, Mary Ann (1993): Building a large annotated corpus of english: The penn treebank, *Computational Linguistics* **19**(2): 313–330.
- Montemurro, Marcelo A. & Zanette, Damián H. (2013): Keywords and co-occurrence patterns in the voynich manuscript: An information-theoretic analysis, *PLoS ONE* **8**(6): e66344.
- Mühling, Andreas, Hubwieser, Peter & Brinda, Torsten (2010): Exploring teachers' attitudes towards object oriented modelling and programming in secondary schools, *Proceedings of the Sixth international workshop on Computing education research*, Aarhus, Denmark, August 9-10 2010, ACM, New York, pp. 59–68.
- Müller, P. & Hubwieser, P. (2000): Informatics as a mandatory subject at secondary schools in bavaria, *Open Classrooms in the Digital Age - Cyberschools, e-learning and the scope of (r)evolution*, EDEN, Budapest, pp. 13–17.
- Nesbit, John C. & Adesope, Olusola O. (2006): Learning with concept and knowledge maps: A meta analysis, *Review of Educational Research* **76**(3): 413–448.
- Nicoll, Gayle (2001): A three-tier system for assessing concept map links: a methodological study, *International Journal of Science Education* **23**(8): 863–875.
- Novak, Joseph D. (2002): Meaningful learning: The essential factor for conceptual change in limited or inappropriate propositional hierarchies leading to empowerment of learners, *Science Education* **86**(4): 548–571.
- Novak, Joseph D. (2010): *Learning, Creating, and Using Knowledge: Concept Maps As Facilitative Tools in Schools and Corporations*, 2nd ed., Routledge, London.

- Novak, Joseph D. & Cañas, Alberto J. (2008): The theory underlying concept maps and how to construct and use them, Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Institute for Human and Machine Cognition, Florida.
- Novak, Joseph D. & Cañas, Alberto J. (2010): The universality and ubiquitousness of concept maps, in J. Sánchez, A. J. Cañas & J. D. Novak (eds), *Concept Maps: Making Learning Meaningful: Proceedings of the Fourth International Conference on Concept Mapping*, Viña del Mar, Chile, October 5-7, 2010, Vol. 1, Universidad de Chile, Chile, pp. 1–13.
- Novak, Joseph D. & Gowin, Bob (1984): *Learning how to learn*, Cambridge University Press, Cambridge and New York.
- Novak, Joseph D. & Musonda, Dismas (1991): A twelve-year longitudinal study of science concept learning, *American Educational Research Journal* **28**(1): 117–153.
- O'Connor, Debra L., Johnson, Tristan E. & Khalil, Mohammed (2004): Measuring team cognition: Concept mapping elicitation as a means of constructing team shared mental models in an applied setting, in A. J. Cañas, J. D. Novak, F. M. González, A. Cañas & F. M. González García (eds), *Concept maps: Theory, methodology, technology: Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, Sept 14-17 2004, Vol. 1, pp. 487–494.
- Ozdemir, A. (2005): Analyzing concept maps as an assessment (evaluation) tool in teaching mathematics, *Journal of Social Sciences* **1**(3): 141–149.
- Özdemir, Gökhan & Clark, Douglas B. (2007): An overview of conceptual change theories, *Eurasia Journal of Mathematics, Science & Technology Education* **3**(4): 351–361.
- Passmore, Graham James (1999): *Concept Maps and the Processes of Comprehension: Explicating Cognition and Metacognition, Structural Knowledge and Procedural Knowledge*, PhD thesis, University of Toronto, Toronto.
- Patil, Kaustubh & Brazdil, Pavel (2007): Sumgraph: Text summarization using centrality in the pathfinder network, *IADIS International Journal on Computer Science and Information Systems* **2**(1): 18–32.
- Piaget, Jean (1929): *The child's conception of the world*, International library of psychology, philosophy, and scientific method, Routledge & K. Paul, London.
- PISA 2009 Technical Report (2012): PISA, OECD Publishing, Paris.

- Pons, Pascal & Latapy, Matthieu (2006): Computing communities in large networks using random walks, *Journal of Graph Algorithms and Applications* **10**(2): 191–218.
- Posner, George J., Strike, Kenneth A., Hewson, Peter W. & Gertzog, William A. (1982): Accommodation of a scientific conception: Toward a theory of conceptual change, *Science Education* **66**(2): 211–227.
- Quirin, A., Cordón, O., Santamaría, J., Vargas-Quesada, B. & Moya-Anegón, F. (2008): A new variant of the pathfinder algorithm to generate large visual science maps in cubic time, *Information Processing & Management* **44**(4): 1611–1623.
- Quirin, Arnaud, Cordón, Oscar, Guerrero-Bote, Vicente P., Vargas-Quesada, Benjamín & Moya-Anegón, Felix (2008): A quick mst-based algorithm to obtain pathfinder networks ( $\infty, n - 1$ ), *Journal of the American Society for Information Science and Technology* **59**(12): 1912–1924.
- Ragonis, Noa & Ben-Ari, Mordechai (2005): A long-term investigation of the comprehension of oop concepts by novices, *Computer Science Education* **15**(3): 203–221.
- Rajaraman, Anand & Ullman, Jeffrey David (2011): *Mining of Massive Datasets*, Cambridge University Press, Cambridge.
- Reichardt, Jörg & Bornholdt, Stefan (2006): Statistical mechanics of community detection, *Physical Review E* **74**(1): 016110.
- Riedl, Alfred (2010): *Grundlagen der Didaktik, Pädagogik*, 2 ed., Steiner, Stuttgart.
- Rosas, Scott R. & Kane, Mary (2012): Quality and rigor of the concept mapping methodology: A pooled study analysis, *Evaluation and Program Planning* **35**(2): 236–245.
- Rousseeuw, Peter J. (1987): Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* **20**: 53–65.
- Ruiz-Primo, Maria Araceli (2004): Examining concept maps as an assessment tool, in A. J. Cañas, J. D. Novak, F. M. Gonzáles, A. Cañas & F. M. González García (eds), *Concept maps: Theory, methodology, technology: Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, Sept 14-17 2004, Vol. 1, pp. 555–562.

- Ruiz-Primo, Maria Araceli, Schultz, Susan E., Li, Min & Shavelson, Richard J. (2001): Comparison of the reliability and validity of scores from two concept-mapping techniques, *Journal of Research in Science Teaching* **38**(2): 260–278.
- Ruiz-Primo, Maria Araceli & Shavelson, Richard J. (1996): Problems and issues in the use of concept maps in science assessment, *Journal of Research in Science Teaching* **33**(6): 569–600.
- Ruiz-Primo, Maria Araceli, Shavelson, Richard J. & Schultz, Susan Elise (1997): On the validity of concept map-base assessment interpretations: An experiment testing the assumption of hierarchical concept maps in science, Technical Report CSE Technical Report 455, Center for the Study of Evaluation (CSE), Graduate School of Education & Information Studies, University of California.
- Rye, James A. & Rubba, Peter A. (1998): An exploration of the concept map as an interview tool to facilitate the externalization of students' understandings about global atmospheric change, *Journal of Research in Science Teaching* **35**(5): 521–546.
- Sabitzer, Barbara (2011): Neurodidactics: Brain-based ideas for ICT and computer science education, *The International Journal of Learning* **18**(2): 167–177.
- Safayeni, Frank, Derbentseva, Natalia & Cañas, Alberto J. (2005): A theoretical note on concepts and the need for cyclic concept maps, *Journal of Research in Science Teaching* **42**(7): 741–766.
- Sanders, Kate, Boustedt, J., Eckerdal, A., McCartney, Robert, Moström, Jan Erik, Thomas, Lynda & Zander, C. (2008): Student understanding of object-oriented programming as expressed in concept maps, *SIGCSE Bulletin* **40**(1): 332–336.
- Sanders, Kate & Thomas, Lynda (2007): Checklists for grading object-oriented cs1 programs: concepts and misconceptions, *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, Dundee, UK, June 25-27, 2007, ACM, New York, pp. 166–170.
- Schöneberg, Hendrik (2010): Context vector classification - term classification with context evaluation, in A. L. N. Fred & J. Filipe (eds), *KDIR 2010 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, Valencia, Spain, October 25-28, 2010, SciTePress, pp. 387–391.
- Schvaneveldt, Roger W., Dearholt, D.W & Durso, F.T (1988): Graph theoretic foundations of pathfinder networks, *Computers & Mathematics with Applications* **15**(4): 337–345.

- Schvaneveldt, Roger W., Durso, Francis T. & Dearholt, Donald W. (1989): Network structures in proximity data, *The Psychology of Learning and Motivation* **24**: 249–284.
- Schwarz, Gideon (1978): Estimating the dimension of a model, *Annals of Statistics* **6**(2): 461–464.
- Schwill, Andreas (1994): Fundamental ideas of computer science, *Bull. European Assoc. for Theoretical Computer Science* **53**: 274–295.
- Seidel, Tina & Prenzel, Manfred (2008): Assessment in large-scale studies, in E. Klieme, D. Leutner & J. Hartig (eds), *Assessment of competencies in educational contexts*, Hogrefe & Huber Publishers, Toronto, pp. 279–304.
- Shaw, Mildred L. G. & Woodward, J. Brian (1990): Modeling expert knowledge, *Knowledge Acquisition* **2**(3): 179–206.
- Shulman, Lee S. (1986): Those who understand: Knowledge growth in teaching, *Educational Researcher* **15**(2): 4–14.
- Solomon, Karen O., Medin, Douglas L. & Lynch, Elizabeth (1999): Concepts do more than categorize, *Trends in cognitive sciences* **3**(3): 99–105.
- Sousa, David A. (2009): *How the brain learns: A multimedia kit for professional development*, 3rd ed., Corwin Press, Thousand Oaks.
- Sowa, John F. (1984): *Conceptual structures: Information processing in mind and machine*, Addison-Wesley, Reading.
- Squire, Larry R. (1987): *Memory and brain*, Oxford University Press, New York.
- Stibor, Thomas (2008): Discriminating self from non-self with finite mixtures of multivariate bernoulli distributions, *Proceedings of the 10th annual conference on Genetic and evolutionary computation (CD-ROM)*, Atlanta, USA, July 12-16 2008, ACM, New York.
- Strautmane, Maija (2012): Concept map-based knowledge assessment tasks and their scoring criteria: An overview, in A. J. Cañas, J. D. Novak & J. Vanhear (eds), *Concept Maps: Theory, Methodology, Technology: Proceedings of the Fifth International Conference on Concept Mapping*, Valletta, Malta, Sept 17-20 2012, Vol. 1, pp. 80–88.
- Strike, K. A. & Posner, G. J. (1992): A revisionist theory of conceptual change, in R. A. Duschl & R. J. Hamilton (eds), *Philosophy of science, cognitive psychology, and educational theory and practice*, State University of New York Press, Albany, pp. 147–174.

- Taricani, Ellen M. & Clariana, Roy B. (2006): A technique for automatically scoring open-ended concept maps, *Educational Technology Research and Development* **54**(1): 65–82.
- Toutanova, Kristina, Klein, Dan, Manning, Christopher D. & Singer, Yoram (2003): Feature-rich part-of-speech tagging with a cyclic dependency network, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Canada, May 27 - June 1 2003, pp. 252–259.
- Trumpower, David L. & Goldsmith, Timothy E. (2004): Structural enhancement of learning, *Contemporary Educational Psychology* **29**(4): 426–446.
- Trumpower, David L., Sharara, Harold & Goldsmith, Timothy E. (2010): Specificity of structural assessment of knowledge, *The Journal of Technology, Learning, and Assessment* **8**(5).
- Truong, Nghi, Roe, Paul & Bancroft, Peter (2004): Static analysis of students' java programs, *Proceedings of the Sixth Australasian Conference on Computing Education - Volume 30, ACE '04*, Australian Computer Society, Inc, Darlinghurst, Australia, pp. 317–325.
- Tversky, Amos (1977): Features of similarity, *Psychological Review* **84**(4): 327–352.
- Valerio, Aljandro, Leake, David B. & Cañas, Alberto J. (2008): Automatic classification of concept maps based on a topological taxonomy and its application to studying features of human-built maps, in A. J. Cañas, P. Reiska, M. Åhlberg & J. D. Novak (eds), *Concept Mapping: Connecting Educators: Proceedings of the Third International Conference on Concept Mapping: Tallinn, Estonia & Helsinki, Finland: September 22-25 2008, Vol. 1*, Tallinn University, Estonia, pp. 122–129.
- Weber, Susanne & Schumann, Matthias (2000): Das Concept Mapping Software Tool (COMASOTO) zur Diagnose strukturellen Wissens, in H. Mandl & F. Fischer (eds), *Wissen sichtbar machen*, Hogrefe, Göttingen, pp. 158–179.
- Weinert, Franz E. (2001): Vergleichende Leistungsmessung in Schulen - eine umstrittene Selbstverständlichkeit, in F. E. Weinert (ed.), *Leistungsmessungen in Schulen*, Beltz Pädagogik, Beltz-Verl., Weinheim and Basel, pp. 17–31.
- Witrock, Merlin C. (1992): Generative learning processes of the brain., *Educational Psychologist* **27**(4): 531–541.
- Wolfe, John H. (1970): Pattern clustering by multivariate mixture analysis, *Multivariate Behavioral Research* **5**: 329–350.



- Yin, Yue, Vanides, Jim, Ruiz-Primo, Maria Araceli, Ayala, Carlos C. & Shavelson, Richard J. (2005): Comparison of two concept-mapping techniques: Implications for scoring, interpretation, and use, *Journal of Research in Science Teaching* **42**(2): 166–184.
- Yoo, Jin Soung & Cho, Moon-Heum (2012): Mining concept maps to understand university students' learning, in K. Yacef, O. Zaïane, H. Hershkovitz, M. Yudelson & J. Stamper (eds), *Proceedings of the 5th International Conference on Educational Data Mining*, Chania, Greece, June 19-21 2012, pp. 184–187.
- Zimmaro, Dawn, Zappe, Stephen M., Parkes, Jay T. & Suen, Hoi K. (1999): Validation of concept maps as a representation of structural knowledge, *Annual Meeting of the American Educational Research Association*, Montreal.