In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Chapter 14
Collaborative Data Management

Sven-Eric Schapke, Jakob Beetz, Markus König, Christian Koch and André Borrmann

Abstract The design, construction and operation of buildings is a collaborative process involving numerous project participants who exchange information on an ongoing basis. Many of their working and communication processes can be significantly improved by using a uniformly structured building information model. A centralized approach to the administration of model information simplifies coordination between project participants and their communications and makes it possible to monitor the integrity of the information as well as to obtain an overview of project progress at any time. Depending on which model information from which project phases and/or sections need to be worked on by which partners, different forms and means of cooperation can be employed. This chapter presents different methodical approaches, practical techniques and available software systems for cooperative data administration. It discusses the different information resources and possible forms of cooperation for model-based collaboration and explains the underlying technical

Sven-Eric Schapke
think project! GmbH, Zamdorfer Str. 100, 81677 Munich, Germany
e-mail: sven-eric.schapke@thinkproject.com

Jakob Beetz
RWTH Aachen University, CAAD – Design Computation, Schinkelstr. 1, 52062 Aachen, Germany
e-mail: beetz@caad.arch.rwth-aachen.de

Markus König
Ruhr-Universität Bochum, Chair of Computing in Engineering, Universitätsstr. 150, 44780 Bochum, Germany
e-mail: koenig@inf.bi.rub.de

Christian Koch
Bauhaus-Universität Weimar, Chair of Intelligent Technical Design, Marienstr. 13a, 99423 Weimar, Germany
e-mail: c.koch@uni-weimar.de

André Borrmann
Technical University of Munich, Chair of Computational Modeling and Simulation, Arcisstr. 21, 80333 Munich, Germany
e-mail: andre.borrmann@tum.de

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

concepts, such as concurrency checking and versioning along with rights and permissions management. Several different software systems available for cooperative data administration are also presented. The chapter concludes with a brief look at future developments and the challenges still to be faced.

14.1 Introduction

The design and construction of buildings is a highly collaborative process involving numerous participants. Clear communications between these participants, which are vital to successfully completing the projects, is still primarily based on 2D drawings today. For complex building projects, this form of information exchange is, however, both time consuming and prone to errors. The monitoring, coordination and agreement of changes is not automatically supported. The introduction of digital building information models with agreed workflows and processes offers an effective way of supporting and improving different forms of collaboration. Information needed from the various project participants can be kept up to date and made immediately available in a shared information space, which can be semi-automatically verified and monitored for inconsistencies. Iterative planning cycles are kept short, project progress is easier to monitor and control, and communications between all participants are more reliable as everyone has access to the same information. The transition to more effective and efficient computer-supported collaborative systems does, however, require fundamental changes to the way we work compared with paper-based work processes.

This chapter shows how collaboration as well as the coordination of planning processes can be significantly improved through cooperative data management. To begin with, we look at fundamental concepts of shared information spaces as well as Computer-Supported Collaborative Work (CSCW). In the first section, we present the basic principles of BIM-based information resources and their processing methods, and in the second, we look at fundamental aspects of cooperative data management.

In the following we present an overview of the different available technologies and their applications. The third section discusses software tools that support the various concepts and methods of collaborative working, the different approaches they take in supporting these concepts, and their respective requirements. Finally, we conclude the chapter with a critical consideration of the current state of the art and take a look at future developments and research in the field.

14.2 BIM Information Resources

The basis of every data management task is clearly addressable and formally uniform amount of data, known as data sets, data objects or *information resources*. Data

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

management systems describe these information resources with the help of metadata to make them easier to capture, organize, find and use. In model-based collaboration processes, first structured information resources, such as object-oriented 3D building models, need to be managed. These can happen at different levels of aggregation, for example at the level of a building element, of groups of elements, or of entire models. At the same time, building projects always also involve a degree of semi-structured information resources, for example text, images or drawings. While these may be created and edited using software applications, they are only interpreted by the user in their respective context.

14.2.1 Metadata

The basis for the consistent organization of information resources is metadata schemas. Traditionally, these outline a series of metadata attributes that represent different aspects of a resource, for example to:

- identify them (e.g. ID, storage address, creator, author),
- describe their content (e.g. application field, level of detail, project area),
- describe their technical properties (e.g. data format, size),
- describe their functional state (e.g. version, revision, work status), and
- retain them for the future (e.g. safety copies, archives, migration).

In cases where individual data objects require very detailed descriptions, including relationships between them within a model, object-oriented meta models may be created and used for the automatic generation of software components (see Tozer, 1999).

The most important metadata attribute of an information resource is its identifier. This is a unique descriptor for identifying a resource that is defined by the resource itself or assigned by the management system. In digital building models, all important elements generally have a GUID (Globally Unique IDentifier) or a UUID (Universally Unique IDentifier). These make it possible to manage each element individually, to compare the same element in different model versions and to reference elements in external software systems. For this to work, all systems in a collaborative process must, however, preserve the GUIDs of the central model, and not, for example, regenerate them when exporting into a neutral data format.

Additional consistency between attributes can be achieved through the use of metadata vocabularies. These define a series of possible attribute values, for example through lists, classifications, partonomies or other classification systems.

The first use of metadata in current construction practice is in drawing management. Within a project, a unified coding system, the drawing code, is used to identify and describe each drawing. The drawing code combines several classification facets that employ a predefined vocabulary, for example for (1) sub-projects, (2) trades (architecture, structural, etc.), (3) forms of presentation (floor plan, section, etc.) and (4) project phase (concept design, design development, etc.) as well as for versions

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

and revisions. As shown in Sect. 14.2.3, such identification codes can also be used to manage digital building models. Metadata vocabularies can follow established standards, for example for general construction classification systems such as OmniClass (2006), Uniclass2 (2013), cost classification systems such as the German DIN 276 (2008) or element catalogs such as the KKS (2010). For more information, see the discussion in Chap. 6.

14.2.2 Level of Aggregation

A vital aspect for the application of a data management system is the level of aggregation of the information resources. In order to immediately find and edit specific project information, each resource should hold only a limited quantity of information. As the number of resources increases, however, so too does the effort required to manage them. In practice, therefore, a lot of project information is managed in an aggregated form before they are read and used by other software systems. Table 14.1 shows five different levels of aggregation and some examples of corresponding information resources and management systems.

Information resources with a high level of aggregation include, for example, collections of models and documents, such as a CAD file that contains a 3D model along with corresponding 2D drawings and a bill of quantities. The aggregated resources act as a container for different kinds of information, which are now more difficult to access as they must first be retrieved, loaded, interpreted and filtered. Information resources with an intermediate level of aggregation comprise related information from individual work tasks and building systems, for example a section of a building, a floor, or a particular assembly of parts. These are often saved in separate files. Information resources with a low level of aggregation (i.e. a higher level of detail) represent individual logical units within a model or document, such as individual building elements, element properties or text segments. Within a model, these datasets are interconnected and must therefore be managed in a common, coordinated system.

14.2.3 Digital Building Models

The basis for model-based collaboration is the digital building models that are created by the respective project participants using a variety of different software tools. These models represent certain domain-specific aspects of a building and are therefore called *domain models*. In planning phases, domain models typically represent specific elements of a building structure or space and their geometric, functional and material properties. In other project phases, these elements can also have conceptual properties, such as deadlines or costs, but may conversely not always hold a 2D or 3D representation of the element.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Table 14.1 A comparison of information resources with different levels of aggregation

Level of aggregation	**Collection, models and documents**	**Individual model, individual document**	**Element group, subset of model**	**Building element**	**Element property**
Example	5D building model CAD project file with model and drawings	3D building model Contract document CAD drawing	Model of the walls of a floor Model of a section of a building 3D marker	Single building element (e.g. a column or room)	Dataset with properties of a building element (e.g. material parameters of a concrete column)
Level of aggregation	high	intermediate		low	
Advantages / Disadvantages	Low number of resources Lower management workload No direct access to detailed information	Medium number of resources Moderate management workload Some access to detailed information		Large number of resources High management workload Direct access to detailed information	
Suitable data management systems	File repository Document management system Internet-based project platform	Document management system Internet-based project platform Product data management system		Product data management system Product model server	

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

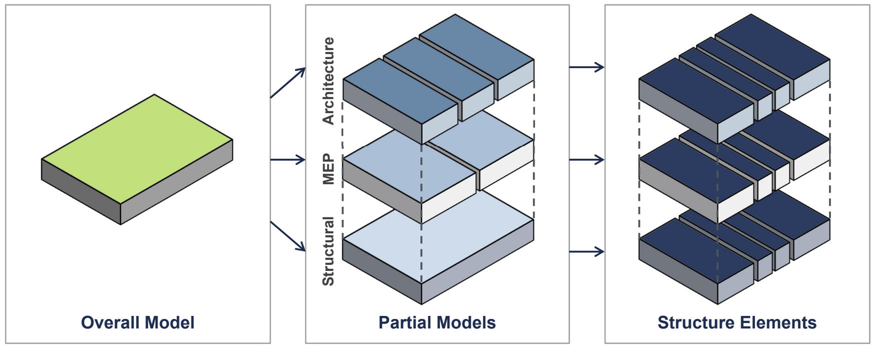

Fig. 14.1 Subdivision of a building into partial models

Over the course of a building project, the project participants create a large number of domain models to design and document the building, its construction and use. Each of these domain models represents a part of the building and its lifecycle and is correspondingly also known as a *partial model*.

For better management, these partial models can be classified according to different aspects. Relevant classification dimensions are in particular the domain, zone, level of detail and project phase.

The *domain* represents the disciplinary perspective and conceptualization of a model. It is primarily defined by the represented technical, functional and economic aspects of a building, the envisaged use of the model and the discipline of the model creator and the software they use. In a project, the classification of domains depends on the kind of building, the project organization of software systems used. The detailed requirements of the domain-specific model content of a particular domain can be defined with the help of Model View Definitions, as described in Chap. 7.

The *zone* of a model specifies the spatial areas that a model encompasses. The classification of zones is in effect a spatial subdivision of a building project, for example into sub-projects, stories, or building sections as is frequently set out in project structure plans. In addition, further detailed compositional structures (partonomies) and topological systems can be used to determine whether a model touches, intersects or contains other models (see Chap.17).

Figure 14.1 illustrates the subdivision of a model into six sub-models with three domains (vertical: architecture, building services, structure) and six logical spatial zones (Overall, east – west, east wing – atrium – west wing) as well as the dividing elements that result from the combination of classification dimensions.

The *level of detail* of a model indicates how precisely the elements of a model represent the specific objects. In building models, the level of detail is initially a factor of which geometric parts of a building element are represented, for example, the frame, the door leaf, and the door handle. Alternatively, a geometric element might be simple but include a detailed description of the respective element. Depending on the focus of the classification, this degree of detail is known either as Level of

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Detail (BS PAS 1192-2:2013) or Level of Development (AIA Document E202) (see Chap. 7).

The *phase* indicates at what time and for what purpose a model was created, and/or which status it currently has. Phase classifications can be based on quite different process structures, for example the overarching project phase and value creation processes, or alternatively the individual work steps and their corresponding processing statuses.

In model management, changes in phases often correspond to changes in the status of the information resources. On the project, statuses must be defined that are reached by requesting, checking, revisioning, filtering, transcoding and linking of a model. At the same time, mechanisms for propagating status changes need to be established that indicate how status changes in individual information resources relay to linked, subordinate and superordinate resources, as well as to resources in adjacent zones and domains.

14.2.4 Information in model coordination and model management

In addition to digital building models, there are a number of other information resources that arise through the coordination and verification of models and through their evaluation and further use. Figure 14.2 shows an example of a coordination model consisting of several partial models, a 3D marker in the coordination model as well as other accompanying documents and drawings.

A coordination model is a model that collects several partial models and serves as a central resource for model-based collaboration. Coordination models can be created for very different purposes and typically have an own author and lifecycle. The primary aim of a coordination model is to check that separately created partial models are consistent with one another and do not exhibit geometric clashes or other kinds of inter-domain conflicts. For the combined checking of multiple building models, different BIM applications, so-called viewers or model checkers can be used (see Chap. 19). Other application possibilities include, for example, comparing versions, variants and actual versus intended model states as well as to 'locate' certain processes and documents in the overall model.

The results of model checking are likewise important information resources. Typically, a manual or (semi-)automatic checking procedure, e.g. from clash detection, will set 3D markers with comments in a coordination models to flag clashes and uncertainties. In collaborative processes, these quality control checks need to be collated into checklists in order to coordinate their execution and/or clarification. Chapter 7 shows how corresponding 3D markers can be saved, exchanged and managed in a neutral BIM Collaboration Format (BCF).

Alongside such checklists, all documents that relate to the building models represent important information resources. These can be drawings and bills of quantities generated from the building models, and may need verifying and updating with each new version of the model. Or they can be independently created documents

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

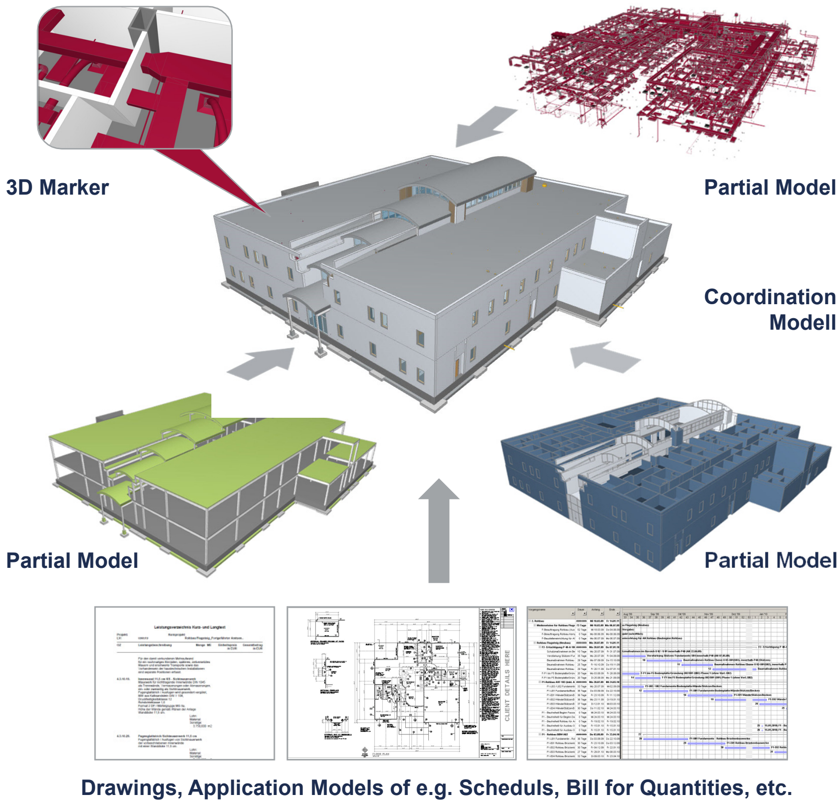

Fig. 14.2 The integration of partial models and documents in a coordination model

and models that are associated with the building model, for example, construction details or delivery schedules that refer to selected model elements. In recent years, research has been undertaken on ways of exchanging and managing combinations of multiple models from 4D and 5D BIM applications as well as corresponding documents with the help of so-called multi models, linked models or linked data techniques (Scherer and Schapke, 2015; Pauwels et al., 2015; Beetz, 2009).

14.3 The Requirements of Cooperative Data Management

A fundamental function of data management is the efficient provision of all information that the different project participants need. In collaborative work, participants need adequate information in order to agree, coordinate and direct their work towards common objectives. At the same time, the amount of information must be manageable in order to able to coordinate and control collaborative processes. Those

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

who commission design and planning services must be able to ensure that important information from the respective project participants is created in the desired quality, properly stored, adequately documented in a verifiable form and efficiently distributed to the respective participants.

The requirements for the management of model-based project information depend to a large degree on the project organization, the complexity of the building project and the software systems used. In addition to defining suitable filing structures and coding systems for different project information, rules must be established for editing and saving information. The following five sections provide an overview of different forms of cooperative data management and the technical processes that support this. The application methods and the requirements that each of these have depends in particular on the following aspects:

- *Communication and Cooperation:* how many project participants, in which locations, at what times, from which organizations, with which contractual relationships work together,
- *Concurrency:* when and to what degree does the integrity of information created in parallel need to be ensured,
- *Roles and Rights:* to what degree is project information confidential and only for certain (groups of) participants, to what degree do participants need to have different editing rights and coordination responsibilities and which creation, access and usage rights result from these,
- *Versioning:* in what detail do individual work steps for editing project information and the resulting changes and variants need to be recorded and reliably documented,
- *Approval and Archiving:* how will certain defined planning stages be secured, definitively stored and published for others.

In document management systems, established methods for data management exist that fulfill these functional requirements. These methods deal with drawings, reports and photographs of a project almost exclusively as distinct data containers, and only rarely consider their contents. A key question for the management of model information is therefore how such methods can be applied to other levels of aggregation, for example to manage access rights or track versions of individual elements and element assemblies.

In principle, conventional data management methods can be applied to all levels of aggregation, however, this is very rarely useful in building practice. On the one hand, many methods, such as concurrency control, require very close interaction between software applications and the central data management system. This is generally only possible when all key project participants work with the same software systems. In building projects, however, a variety of specialist software systems are typically used (e.g. CAD, CAE, ERP) producing and using different forms of documents, drawings, models and other media data.

Moreover, collaborative work on project data results in a large number of dependencies over different levels of aggregation. These need to be taken into account by data management systems to avoid inconsistencies and conflicts. For example,

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

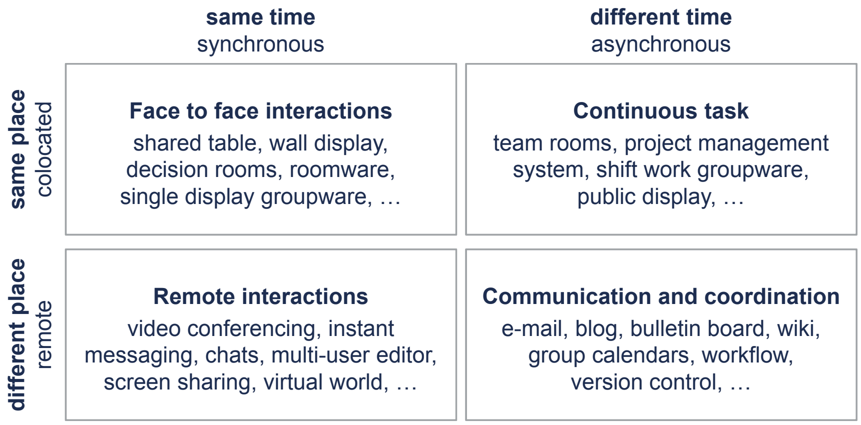

Fig. 14.3 Time-space matrix of communication forms (after Johansen, 1988)

access rights to an element assembly must also determine whether this also confers the right to edit the individual constituent elements, and whether changes made to these also changes the ownership and version of the edited elements, or of the entire element assembly. Many such dependencies can be regulated in detailed, for example through inheritance rules. The high technical complexity required to ensure the conflict-free management of dependencies and the extra work involved and/or the restrictions these entail for users are often disproportionate to the benefits of such comprehensive change control management.

In practice, therefore, one needs to decide what cooperative approaches and data management methods are appropriate for the respective application area. The aim is to find a good balance between a technically simple and user-friendly approach and a data management system that ensures the integrity, reliability and authenticity of all project data at all times.

14.4 Communication and Cooperation

A prerequisite for collaboration is efficient communication between the cooperating partners. The communications medium and the kind of communications influences the form and quality of collaboration.

Depending on the spatial and temporal distribution of the participants, communications can be *synchronous* or *asynchronous* as well as *co-located* or *remote*. The combination of these classifications results in four different communication forms, usually depicted in a time-space matrix as shown in Fig. 14.3 (Johansen, 1988).

At the same time, information can be exchanged through *direct* or *indirect* communications (see Fig. 14.4). In direct communications, partners send information and messages to one another directly. In indirect communications, information is

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

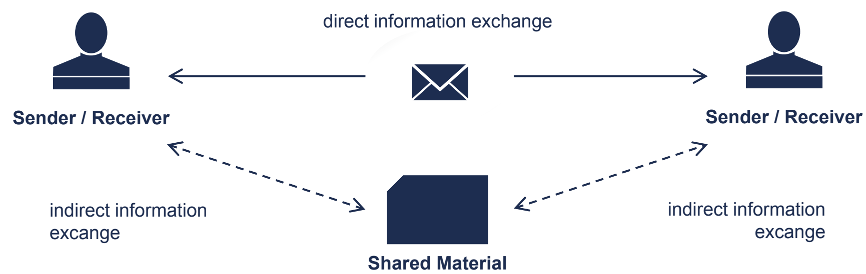

Fig. 14.4 Collaboration (after Schrage, 1990)

exchanged indirectly when working jointly on shared information resources. Shared information resources make it possible to collect relevant communications in a central location so that the solutions chosen and decisions made are transparent and can be understood.

In the design of model-based collaboration, also the *information delivery processes* must be examined in order to determine which participants require which information, when and from whom, as well as which information they create and need to provide to which other participants. The objective is to design and adjust work processes and technical interfaces to one another as seamlessly as possible to ensure the continuous and efficient use of information (see also Chaps. 4 and 7).

Compared with other industries with stationary facilities, the project organization of building projects presents particular challenges for process management and process optimization. In each project, a functioning distribution and delivery network needs to be established in a short space of time in which changing project partners can be incorporated into cross-enterprise processes and at the same time have the opportunity to further optimize their own internal business processes (Bøllingtoft et al., 2011). In addition, many of these fragmented processes involve interdisciplinary and iterative planning tasks.

In *cross-enterprise collaboration*, as well as in interdisciplinary planning teams, it is likewise important, aside from sharing information, to consider economic aspects, such as the effective coordination and control of project partners as well as their property and usage rights, the preparation and correct legal documentation of decisions or the balance of group dynamics in interdisciplinary teams. Depending on which of these aspects plays a central role, communications and collaboration can take different forms. The following terms describe these different forms:

- *Communication* describes exchanges of information between two or more human, technical or institutional participants in the form of messages.
- *Interaction* describes reciprocal communicative activities by people who through their actions want to achieve certain effects among other people.
- *Coordination* describes interactions that are necessary to achieve the efficient and effective alignment of the targeted activities of several people.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

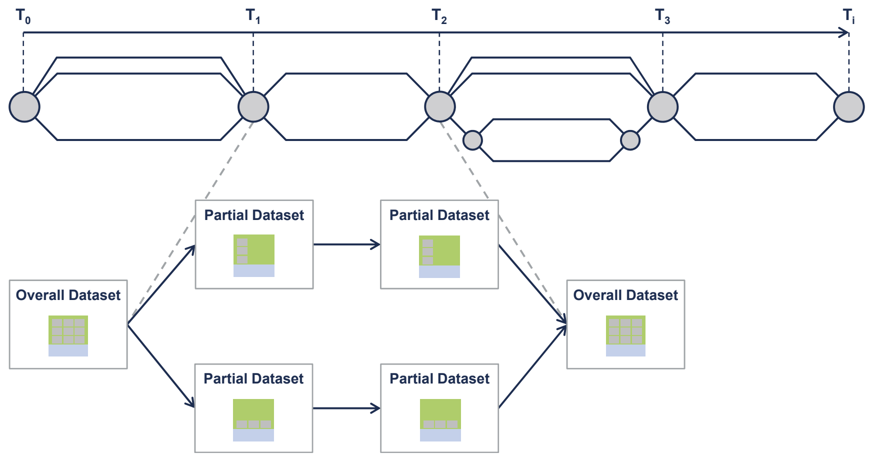

Fig. 14.5 The coordination of individual planning stages through the merging of partial datasets

- *Cooperation* is the working together of several participants on joint material to achieve common aims. Cooperation is typically voluntary and trust-based, which promotes clear communications and effective coordination.
- *Collaboration* describes the cooperation of complementary partners with a high level of trust and reciprocal support. An important objective of collaboration is the creation of collective knowledge in order to develop solutions for complex problems. Collaborative processes are frequently highly creative, and all partners are of equal standing.

14.4.1 Concurrency Control

For data management in model-based collaborative environments, the distributed and synchronous editing of shared information resources presents a particular problem. In practice, several project participants often work concurrently on their respective copies of a building model or document that may have been created by another project participant. Changes made to this copy can lead to technical inconsistencies and disciplinary conflicts in the project information because cooperating partners may make decisions based on different assumptions.

Concurrency control offers a way of avoiding inconsistencies or conflicts when working simultaneously on project information. There are two primary approaches to this: *pessimistic concurrency control* avoids conflicts in advance by allowing only certain changes to be made, while *optimistic concurrency control* identifies conflicts in project information and attempts to resolve them after they have occurred.

Figure 14.5 shows a model of distributed synchronous data processing that offers a good basis for discussing different forms of cooperation and concurrency control.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

The model divides the process of cooperation into discrete phases. Each phase ends with a coordination point Ti at which the overall dataset should be consistent and free of conflicts. Within each phase, users undertake a series of work steps in order to:

- extract a partial dataset with the information they require for the specific task from the overall dataset (extraction),
- make their respective changes and additions in their local dataset (modification), and
- feed back the changes from their local dataset to the overall dataset and merge this with other, likewise potentially modified, partial datasets (integration).

These three work steps may be undertaken in parallel by several users. At each coordination point at the end of each phase, the overall dataset must be in a conflict-free state.

Using *pessimistic concurrency control* each partial dataset that is extracted is locked in the overall dataset. The lock prevents other users from working on the same information at the same time and is only removed once the partial dataset has been integrated back into the overall dataset. Pessimistic concurrency control is mostly used in documents and product data management systems as well as in model servers. The ability to work on project information concurrently depends largely on the level of aggregation of the information resources and the extent of each lock, i.e. whether an entire model is locked when information is extracted or whether just a single layer, element or property is locked, making it possible for others to work on other parts of the model in the meantime.

Optimistic concurrency control initially assumes that inconsistencies will arise when local partial datasets are worked on in parallel. The resulting conflicts will therefore need resolving when re-integrating the modified datasets into the overall dataset. This approach is called optimistic because the assumption is that only a few conflicts will arise, and that the effort required to resolve them is reasonable. Optimistic concurrency control is commonly used in software development because changes are made to individual lines of source code and information therefore has a low level of aggregation. As such, this method is the basis of many code management and software configuration management systems such as CVS or GIT.

Both forms of concurrency control have advantages and disadvantages. The pessimistic approach avoids the occurrence of conflicts arising during concurrent work, but also requires that users wait until the relevant (part of a) model or document is unlocked, which can be lengthy. Depending on the level of aggregation, the management of lock releases can quickly become complex and time-consuming. The optimistic approach offers greater freedom in the editing of data, but the management of partial datasets can become extremely complex, especially when merging partial datasets back into an integrated model. For example, integrating several building models entail comparing not just the respective geometric bodies but also the properties of all relevant building elements. Comprehensive strategies are required, for example, to transfer geometric modifications to a changed, merged or deleted element to other models while simultaneously taking into account all dependencies

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

with other elements (for example a ceiling slab that rest on the element) and element data (for example its volume).

In practice, the advantages and disadvantages of the different concurrency control methods must be evaluated for the respective application scenario. Approaches using pessimistic concurrency control are commonly used within in-house systems while optimistic concurrency control dominates in cross-enterprise collaboration scenarios. In addition to all of this, seamless data processing requires that project participants establish rules for how they work together, i.e. by defining clear areas of responsibility for the respective participants.

14.4.2 Roles and rights

The underlying principles of interoperability in data management explained in Chap. 5 refer predominantly to the technical aspects of information modeling. Collaborative approaches to planning and engineering tasks in the construction sector using electronic data requires that we rethink established methods and conventional work practices. With paper-based and document-based forms of collaboration, it is usually clear who is responsible for all facets of information, for example, whoever last stamped or signed the respective planning document (drawing, text, etc.). This is not so straightforward when working on jointly produced and integrated building models, where the kind of ownership, rights and reliability of input need to be considered. For this, project-specific and/or sector-wide agreements have to be reached on how these will be handled.

The authorship of information within one and the same information space (the model) can change from element to element and even from attribute to attribute. Metadata attached to the respective information resource must therefore record who is responsible for which information in each case. The owner of a resource may, for example, be the original creator (e.g. the architect who created the wall element) or alternatively the last person to edit it (e.g. the structural engineer who added reinforcement bars). The ability to edit individual aspects of an element must also be clarified: is the structural engineer allowed, for example, to enter the concrete class as a material attribute of the wall element created by the architect? Or should he create an independent material element and link this to the corresponding wall? Or perhaps create an independent element of his own? The respective rights for reading, writing and deleting can be assigned not just to a specific individual user but also to groups of users, who have different roles, for example all members of a company, a division, all system administrators or the respective project manager. Many common systems allow one to define hierarchical and cascading systems of roles and rights at different levels of aggregation, although this can quickly become more complicated when these overlap (e.g. one user with several roles). As such they must be devised with great care and be regularly monitored. When collaboratively creating information resources and models, new forms of ownership, rights and reliability must be considered, and corresponding agreements met on who can do what. The resulting

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

legal aspects fur such digital collaborative processes – for example, who assumes liability for what – have not yet been fully resolved.

14.4.3 Versioning

In model-based collaboration, different information resources are added, modified, extended or deleted by various people. Every information resource can therefore exist in different *versions*.

All changes made to an information resource should be identifiable through a clear version identifier, for example a version number. In addition to updating the version identifier, every change should also register the name of the editor and time of the change along with further metadata such as associated comments or the software version used. Consecutive changes to an information resource over time result in a history of changes which can be presented as a version graph. Which versions of a resource need to be saved as an instance that will remain available in future, must be determined for each respective application.

A special form of versioning is revisioning. A *revision* collects together a series of versions of information resources once a certain work stage has been reached. This might the case, for example, when a document has been checked by several people and flagged for release, or when models that have been worked on in parallel have been merged. If an information resource requires additions or corrections before it can be released, a new revision maybe be generated for this purpose. A coordination model can, therefore, be seen as a revision of an overall model. Information resources that belong to a defined revision are usually archived in a non-revisable state and can then made available for release.

A further important aspect of model-supported collaboration is the use of variants or branches. If an existing information resource is worked on by two different people or systems, and the changes result in two possible alternative results, for example of a construction detail, one can call this a variant. Variants can be developed independently in parallel and the results may not always be compatible. Over the course of a project, one variant is usually chosen and carried forward. Variants are also important for comparing, evaluating and discussing different possible solutions. Variants are likewise given specific identifiers. In the version graph, variants are shown as branches, and when a branch is incorporated into the main model, the branch merges back into the main version history. Figure 14.6 presents the relationship between version, revision and variants using a simple version graph.

The granularity of versioning corresponds to the respective level of aggregation of the information resource (see Sect. 14.2.2). As such, a version graph can be created for all kinds of information resources: there are versions, revisions and variants of entire model and document collections, of individual models and documents, of element groups, of elements and also of individual element properties.

For versioning files, file naming conventions, systems for managing versions of files and directories (e.g. Concurrent Version Systems, CVS, or Subversion, SVN or

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

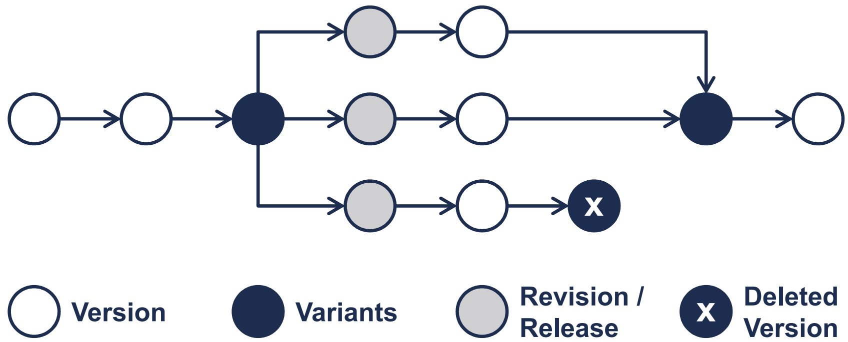

Fig. 14.6 Version graph of an information resource with simple versions (*white*), revision (*dark grey*), variants (*light grey*) and discarded variants (*black with a cross*).

GIT) or document management systems can be used (see Sect. 14.4.2). The simplest approach is to use file names. This means the version, revision and variant can be identified through a part of its filename (e.g. V11_R2_A1 for Version 11, Revision 2 and Alternative 1). Every change is saved as a new file with a new name. The creation of variants using files means that all content must be copied. For building models, this means that the model elements it contains have the same identifiers. This can be useful, for example, when a wall is shifted, and the two variants should be compared. In some cases, however, it can lead to inconsistencies, for example when references are made to model files in another variant. Programs for managing the version of files and directories are generally very good for text files (e.g. Unicode data). When adding a new file, a version number is automatically assigned. Certain stages of progress can be given a revision number. Version management systems also make it possible to track different variants. Such systems are less well suited for binary data as usually each entire variant has to be copied, requiring considerable time as well as storage space. For text-based formats, only the differences between versions and variants need to be saved and transferred. In document management systems, the management of files is usually coupled with the recording of additional metadata in a database. This metadata makes it possible to search for documents using additional information fields. As a rule, these systems do not, however, support variants.

The management of versions of digital building models or partial models is a challenging task. One possibility is to version the entire model file, which for large projects can be several gigabytes in size (a highly aggregated information resource). Most of the time, however, only individual objects within a model change, along with their properties. A selective approach to versioning is, therefore, more advisable, for example at the level of element groups or elements. In most cases, a corresponding database is used that makes it possible to lock, check out and check in individual elements and supports the assignment of specific editing rights. A number of different systems already exist, some of which are proprietary and some open.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

With proprietary systems, access to the model requires the use of special software. For distributed collaborative work processes, all relevant users must use this software. Such systems are available from almost all large BIM software vendors. Support for variants are also partially supported, but not all versions of an element are always saved. In mechanical engineering, for example, Product Data Management (PDM) systems are commonly used (see Sect. 14.5.4). Open systems are typically model servers that make it possible to update partial models that may have been constructed or edited using different software tools. These are based on data standards, such as the Industry Foundation Classes. See Sects. 14.5.5 and 14.5.6 for further information on model servers.

14.4.4 Approval and Archiving

Approval is the process in which an agreed (and often also revisioned) information resource is signed by an authorized partner, published and released for use by others. For example, the approval and release of a construction drawing is signed by the architect. Through this signature, the architect indicates that the drawing is the current and valid basis for planning and construction.

For digital information resources, approval can be indicated by a digital signature. When using files from a common data repository (Sect. 14.5.1), the digital signature denotes the author of the document as well as the definitive status, which should then no longer be changed. Subsequent non-authorized changes are immediately identifiable. Document management systems and internet-based project platforms typically also provide means for defining approval and release processes as well as the use of digital signatures (see Sects. 14.5.2 and 14.5.3). If digital building models are used, a release usually takes the form of a coordination model. Individual domain models can, however, also be approved for release. In such cases, these are typically models that can be saved as distinct files and therefore digitally signed. In practice, printed drawing output from digital building models are still often used and then manually signed. In this case, the relevant state of the building model should also be released, and the corresponding state archived. In current product model servers or BIM model servers, release and archiving processes are currently in development (see Sect. 14.5.5). When using such centralized systems, the model should be saved, digitally signed and released in a standardized data format, for example IFC. This functionality is likely to be implemented in the future.

The storage of data, also over a long period of time, represents a challenge for both technical systems and their users. It is not only imperative that the data can still be read at a later date, but also that it can be re-used. Rapid technological advances in the field of BIM and its associated software products means that requirements for saving and archiving data are growing. The long period of use of buildings and the long guarantee periods for building works is many times longer than the 'half-life' of the data and tools that describe them. A significant problem in this respect is the dependency on proprietary (vendor-specific), closed and insufficiently documented

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

data formats, which has earned the name 'digital amnesia': when products and/or vendors disappear from the market and/or when future versions are no longer compatible with earlier versions of the same software, operating systems or hardware, the data becomes irretrievable. Awareness is growing in the building industry, and in other engineering sectors, of the need for vendor-neutral, self-documenting formats saved in pure text formats with support across numerous sectors, such as STEP or XML (see Chap. 6), as a means of stopping the demise of digital data or at least prolonging its availability. This is particularly relevant for long-term digital archiving for private and public clients who will need to use this data for many years to come.

14.5 Software Systems for Collaborative Work using BIM Data

The various forms and methods of collaboration discussed above can be implemented today using a variety of different technologies and software tools. In this section, we will discuss the main categories, how they work and what applications they are suitable for.

14.5.1 Common file repository

In recent years, common file repositories have developed independently of their respective areas of application into a natural means of organizing collaborative work. They are usually used as part of centralized client-server system architectures as used, for example, within in-house intranets. Simple and traditional implementations can use various protocols to connect to networked drives, FTP servers or Network Attached Storage (NAS) devices. These all require a degree of administration in the setting up of addresses and assignment of rights, and are typically used by users much like external hard drives.

More modern solutions are able to automatically synchronize defined directories and can also incorporate simple mechanisms for versioning, archiving and restoring information resources (e.g. ownCloud, Sharepoint, etc.). Various free and paid services also exist that require no special setup (Dropbox, Google Drive, MS Onedrive, etc.). Here, small firms no longer need to install and administer their own servers, however, they must often agree to sometimes questionable privacy policies that may compromise data privacy and security. A further alternative to server-based solutions are so-called decentralized peer-to-peer networks that synchronize certain files between individual computers (BitTorrent Sync, for example). At present, these are used rarely in practice.

Common to all these simple forms of joint information management is that they offer no specific support for domains, for example the content-based administration of models, drawings and other documents. The most obvious advantage of such systems is their simplicity and ease of use.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

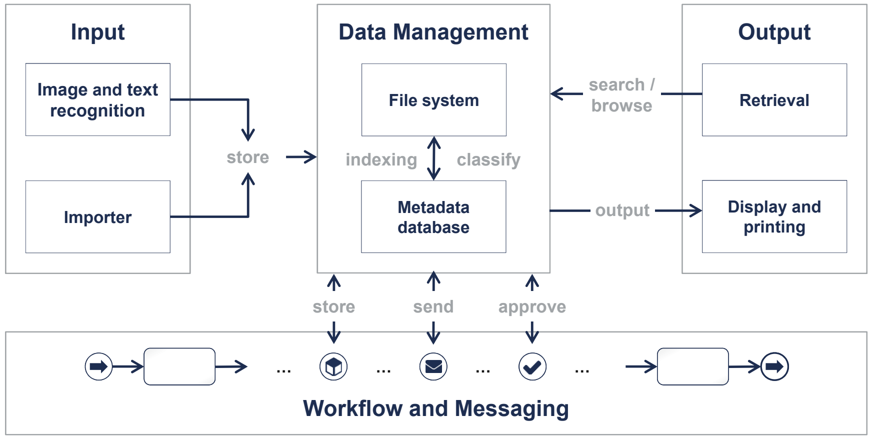

Fig. 14.7 Components of a document management system

14.5.2 Document Management Systems

Document management systems (DMS) have been used in business environments since the 1990s. They offer a central data repository for digital documents and provide various administrative, search and distribution facilities for use in company activities and decision-making processes (Götz et al. 2004).

In DMSs, the documents are treated as data containers whose content can be used by other kinds of software applications. Figure 14.7 shows the typical components of a DMS. The core element of a DMS is its data management system comprising a file repository for documents and a database for their metadata. The consistent description of the document contents, their formalization, creation and use contexts are defined by predefined metadata schemes and vocabularies. Document management is augmented by a user and rights management facility as well as input and output modules for filing and retrieving the documents. Most DMSs provides several modules for the distribution of documents, for example via messages and workflows.

DMSs are often used in conjunction with end user or web apps, either directly or via corresponding connectors with other software systems. The parallel editing of documents is generally managed using a pessimistic concurrency control system. Once a document has been opened (i.e. checked out), it is locked, and other users can only view it but not edit it. Once any changes have been made and the document is saved, it is checked in as a new version in the DMS and is made available to other users to open and edit along with its versioning history.

DMSs are also used in some enterprises to manage BIM data as a means of implementing internal company policies for saving, distribution, release and secure storage of plans and documents for use with BIM models.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

14.5.3 Internet-based Project Platforms

Internet-based project platforms provided a means of centralized information management and the organized filing and distribution of information in cross-enterprise collaboration scenarios. They are also known as collaboration platforms, project spaces or project communication and management systems (PCMS). What makes project platforms appealing is that they are accessible anywhere over the web, flexible enough to be adapted to different application scenarios and that they offer secure data storage and open data connectivity to other software systems.

The basis of any internet-based project platform is a DMS. In contrast to DMSs for in-house use, however, they are often managed by external service providers and offered as a Software as a Service (SaaS). While the implementation of a DMS in a company can entail considerable configuration and operation costs, the project platforms are immediately usable systems pre-configured for cross-enterprise collaboration scenarios. Compared with in-house DMSs, they do not have as comprehensive workflow options and documents are typically not locked to prevent parallel editing.

Internet-based project platforms are used in many construction projects in order to provide all participants with the necessary planning and controlling information. A series of service providers has emerged with services tailored to construction workflows and plan and document management, for example ASite, Aconex, Conject, McLaren and think project!.

Project platforms are first accessed via a web app in a browser, but often provide connectors for typical software applications such as office systems, ERP and CAD systems. In addition they often provide cloud services for processing and evaluating data, for example text recognition, encryption or reporting, along with mobile applications for capturing field data (e.g. defects, photos, construction progress).

Most collaboration providers now also offer special modules for exchanging and using BIM data. Typically, they offer a browser-based 3D viewer, which enables all project participants to visualize and annotate building models without needing any special BIM software. In addition, various means of integrating building models into the collaboration processes are made available. These include centralized versioning (and revisioning) of the models and their combination in the form of coordination models. In addition, models can be linked with 3D markers, drawings and reports in order to process model checklists and conflicts and to monitor dependencies with related document-based communication.

Specialized project platform focused on managing, coordinating and checking building models have also been offered for several years by the larger software vendors (e.g. Autodesk 360 Cloud Services, Nemetschek bim+, GRAPHISOFT BIMcloud) as well as some new specialist providers (e.g. Catenda bimsync). Compared with full-blown BIM solutions, these platform providers allow CAD vendors to work closely with their respective BIM software applications. However, their support for the integration of drawings, documents and project communications is at present only limited.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

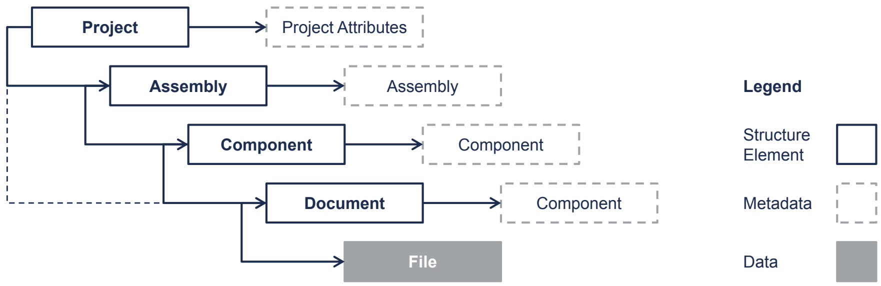

Fig. 14.8 Hierarchical structure of metadata and files in product data management systems

14.5.4 Product Data Management Systems

Product Data Management systems (PDM systems) are software systems for managing product-related information based on a DMS (Schorr et al., 2011). They are most commonly used in industries with stationary facilities, such as in aerospace or automotive manufacturing, shipbuilding and plant engineering.

In a PDM system, the documents are first organized in a structure that corresponds to the product being constructed. The basis is a compositional structure that describes the product in its individual structural elements, i.e. the component assemblies and their components or sub-assemblies. Documents can be linked to all structural elements and described using additional metadata (document attributes). Figure 14.8 shows an example of the hierarchical organizational structure of a PDM in projects, assemblies, components and documents.

All structural elements are saved in individual files and can be annotated with further feature attributes. Figure 14.9 shows a CAD drawing of a wheel assembly. While the element files describe the geometry of the components, information on the assembly of the individual components is stored in an assembly file. In modern 3D CAD systems for mechanical engineering (MCAD systems) this is a typical way of breaking down all the model information. In principle this modeling method can also be achieved using CAD systems for the building sector (AEC CAD systems) such as Autodesk Revit or Autodesk AutoCAD using external references, or XRefs.

Through inheritance mechanisms, the hierarchical product structure can be used to very efficiently annotate elements and documents with keywords and to regulate access rights. To edit the product information, individual elements and the sub-resources can be locked in a top-down cascade and after checking be approved from the bottom up. To support this work approach in so-called Product Lifecycle Management (PLM) processes, systems for supporting such workflows, so-called workflow engines, are employed that facilitate the controlled development of a product using change requests.

A prerequisite for using a PDM system is the use of CAD systems with which product structures can be defined and contextually visualized. Through their ge-

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

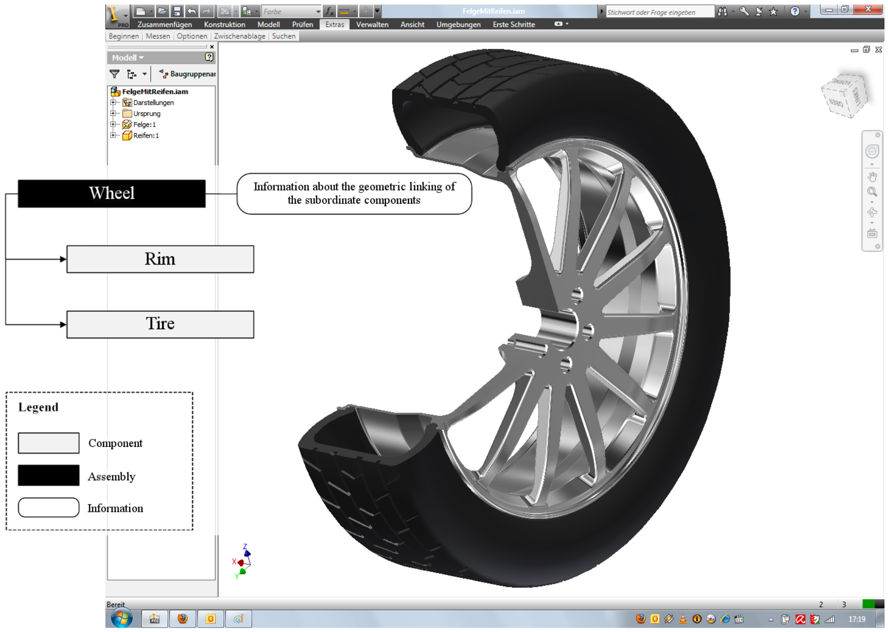

Fig. 14.9 CAD assembly for a wheel comprising the individual components rim and tire

ometric representations, the structural elements, their relationships and respective feature attributes and the attributes of the associated documents can be presented.

14.5.5 Proprietary BIM Servers

The vendors of BIM authoring tools also offer server systems for collaborative model creation. Examples of such vendor-specific systems include the Autodesk Revit Server and Graphisoft BIM Server. These operate with the corresponding software to make partial models available over local (intranet, LAN) or global (Internet, WAN) networks.

The requisite software tools are usually seamlessly integrated into the respective system's authoring tools. Objects can, for example, be edited directly from the user interface of the modeling tool, annotated or locked to prevent further editing. The changes made are saved to the central model on an external server. In ideal cases, the system only saves the differences (so-called deltas) between individual elements, or transfers only attributes. This rapidly improves the editing speed, reduces the data storage requirements and also makes versioning possible.

Other users are then informed of changes made to the model when they open it, or sometimes while they are concurrently working on the model. The greatest limitation of these platforms is that they can only be used with the respective

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

proprietary vendor-specific models which limits or even precludes their availability to other professional disciplines. As with the product models themselves, the network architecture, data exchange protocols and messaging mechanisms are often also vendor-specific and cannot be exchanged with other software applications. For the more comprehensive integration of different software tools and their respective sub-models, product model servers are necessary that are based on vendor-neutral, i.e. standardized product models, data interfaces and communication protocols, as discussed in the next section.

14.5.6 Product Model Servers

Product model servers offer a central management point for product models created by distributed CAD or BIM applications. In contrast to PDM systems, the model data of the elements and their constituent parts are not distributed across several files but are processed in their entirety by the model server and stored in a database (Schorr et al., 2011)(Schorr et al. 2011).

The basis for this database is typically an object-oriented data schema that partially reflects the data format used. To enable the support of different BIM applications, standardized and vendor-neutral product data models, such as the IFC data model, are used, or alternatively generic metadata models such as EXPRESS that also encompass specifications for many other product data standards (see also Chaps. 3, 5 and 6).

To use product model servers, a complete product model must first be created and saved on the server. It is then possible to directly access the individual model elements, the properties and relationships via a corresponding interface. Examples of commercial software packages that can in principle be adapted to work with almost any STEP-based product data model include 'Eurostep Share-A-Space' and the 'Jotne EDM model server', which is commonly used in the processing, automotive and armament industries. Because their architecture is independent of a specific schema, they can in principle be adapted to match any domain-specific model, such as the IFC. This flexibility, however, also entails a high degree of preparatory effort in setting up and using the server. For hierarchical and long-term forms of collaboration and processes such as the development of a new vehicle for mass production, tailor-made solutions can be worthwhile. For the specific requirements and workflows of the construction industry, however, domain-specific solutions are required that are adapted to the recurring processes in the sector and to the corresponding data structures but can also be easily adapted to the specific conditions of each respective project.

Figure 14.10 shows the components of the open-source, freely-available product model server bimserver.org (Beetz et al., 2010). The heart of the data management aspect of the platform is a modeling core that can represent the IFC classification scheme and corresponding instances of a concrete model in a vendor-neutral UML model, and can save this in a configurable database. In addition to extensive inter-

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

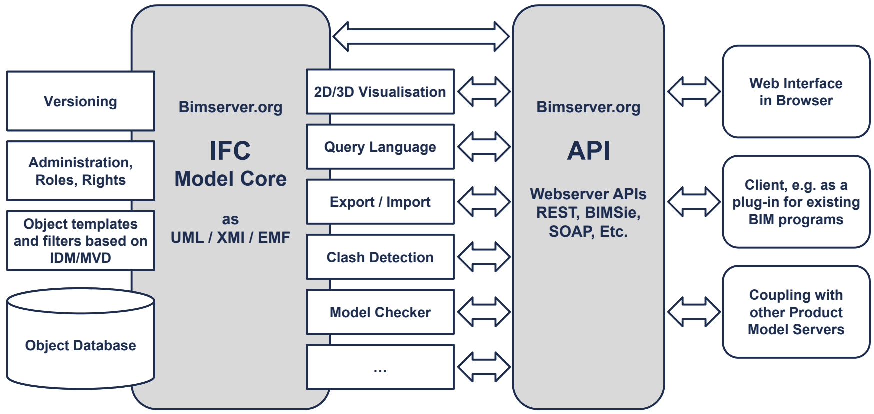

Fig. 14.10 Components of the IFC Product Model Server BIMserver.org

faces for end user applications and a database, the server includes various modules for comparing, validating, filtering, searching, rendering and merging different partial models. These sub-models do not necessarily have to be saved on a single, central server and database instance but can also be switched using vendor-independent protocols (BIM Service interface exchange BIM-Sie, NBIS 2014) between multiple satellite servers in peer-to-peer group constellations. Using such server constellations, for example, individual partial models can be retained and managed by the respective editor and made available, i.e. be 'published' for external project participants. Specific individual applications such as clash detection or model validation based on mvdXML standards (see Chap. 7) can be outsourced as modular services and automatically run each time the model changes.

For the editing of model data, an optimistic cooperation strategy is typically used. Unlike PDM systems, individual elements or documents are not loaded but instead copies of entire partial models are used in order to undertake specific tasks, such as the structural planning. The partial models are not necessarily constructed out of individual assemblies in a product structure but can contain a range of different element data that is not manipulated directly but is required to undertake the respective task.

In addition, because the different project participants work on their partial model for extended periods in parallel (long transactions), the loaded model elements cannot be locked when being worked on. An important task of the model server is therefore to help users identify model changes and conflicts, and in turn to resolve them for integrating the different models (Weise et al. 2004).

Aside from the optimistic concurrency control with long transactions, model servers can, in principle, support distributed synchronous editing of the models when parallel changes to a model are simultaneously transferred and evaluated. The numerous changes made to individual objects, and their corresponding temporary

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

character and real-time synchronization, can, however, quickly lead to an undesirable flood of changes.

14.6 Summary

The methods and techniques for common data management presented in this chapter are some of the most promising but also most complex aspects of Building Information Modeling. There is no simple answer to which of these forms of organizing collaborative work is best; it depends largely on the application scenario in question (e.g. in-house or cross-enterprise), the service phases, the business culture and technical boundary conditions. Although first approaches are already being implemented in practice, as shown in the case studies in part five of this book, there are still a number of questions and areas currently under investigation in research and development. For example, while *legal aspects* of safety, *guarantee* and *liability* are comparatively easily identified in traditional paper-based work processes, practical solutions and strategies are still being sought in the context of predominantly digital collaboration. Digital signatures and fingerprints, archiving strategies and security concepts with respect to roles and rights need considerable further thought and development. Approaches and procedures from other industrial sectors, for example *Product Lifecycle Management* (PLM) or the *Systems Engineering* approaches used in the aerospace industry, offer many interesting concepts that could potentially play a role in defining the future form of computer-supported collaboration in the construction field. For this, however, these methods must be adapted to the multiple small and diverse kinds of project partners in the construction industry.

Today, project participants still predominantly work on shared data from their local desktop computers and workshops in their respective offices. To a certain extent, this due to the high graphics processing power required for working with complex three-dimensional models. Different applications such as the management of shared building model data, the dynamic extension of storage space or the purely numeric computations required for simulations could be distributed across several machines or outsourced to external applications. The management of these distributed and outsourced applications is broadly covered by the heading *cloud computing*. In many cases, however, this software (data management, simulation, etc.) does not need to run on the desktop computer of the user, but could be made available as a networked service (*Software as a Service*). The internet browser will then assume an increasingly important role as a universal graphical user interface.

While there are significant advantages to common data management and processing in the cloud, this is not without some key disadvantages: firstly, cloud service providers must ensure that data belonging to businesses is stored securely and in the long term (e.g. through encryption and archiving) and must be available year-round on the internet. Secondly, many of the cloud applications that run in a web browser, such as those with high graphic processing requirements, are typically not as powerful as the corresponding desktop application. Thirdly, as the number of spe-

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

cific cloud solutions increases, how can project and business data stored on many different servers and services be combined, made available and evaluated?

Technical solutions for the integration of software applications also offer relatively new techniques for the *semantic web* and *linked data* (cf. Chap. 11). Every object and every attribute of a building model can be identified with an URL and stored for linking to one another and later use, with a minimal level of aggregation. The resulting models are directed graphs that are dynamically assembled and can be extended and linked with a range of other information resources, for example sensor date from buildings in the *Internet of Things*. In networked environments, such as for computer-supported collaborative work using building information models, this opens up new kinds of possibilities, which in turn present their own challenges for future research and development.

References

AIA (2013). *AIA Document E202 – 2013, Building Information Modeling Protocol Form*. American Institute of Architects, Washington DC, USA. Retrieved from http://www.aia.org/aiaucmp/groups/aia/documents/pdf/aiab099086.pdf (Accessed November 2014).

Beetz, J., van Berlo, L., de Laat, R., & van den Helm, P. (2010). Bimserver.org – An Open Source IFC Model Server. In: *Proceedings of the CIB W78 2010: 27th International Conference. CIB W78*. Cairo, Egypt, pp. 325–332.

Beetz, J. (2009). *Facilitating distributed collaboration in the AEC/FM sector using Semantic Web Technologies*. Dissertation, Eindhoven University of Technology.

Bøllingtoft, A., Donaldson, L., Huber, G.P., Håkonsson, D.D., & Snow, C.C. (2011). *Collaborative Communities of Firms: Purpose, Process, and Design*. Information and Organization Design Series. Springer New York.

PAS 1192-2 (2013). *Specification for information management for the capital/delivery phase of construction projects using building information modelling*. The British Standards Institution, London, UK.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., & O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. *Advanced Engineering Informatics, 27*(2), pp. 206–219.

DIN 276 (2008). *Kosten im Hochbau*, Beuth, Berlin, Germany

McGrath, J. E. (1984). *Groups. Interaction and Performance*. Englewood Cliffs, NJ, Prentice-Hall.

KKS (2010). *KKS Kraftwerk-Kennzeichensystem*. VGB PowerTech Servic, Essen.

OmniClass (2006). *Construction Specifications Institute (CSI) and Construction Specifications Canada (CSC)*. Alexandria, VA, USA Verlag, Berlin.

Pauwels, P., Tormä S., Beetz, J., Weise, M., & Liebich, T. (2015). Linked Data in Architecture and Construction. *Automation in Construction, 57*, pp. 175–177.

Scherer R.J., & Schapke S.-E. (2015). "A distributed multi-model-based Management Information System for simulation and decision-making on construction projects". *Advanced Engineering Informatics, 25*(4), pp. 582–599. https://doi.org/10.1016/j.aei.2011.08.007

Schorr, M., Borrmann, A., Obergriesser, M., Ji, Y., Gnthner, W., Euringer, T., & Rank, E. (2011). Using Product Data Management Systems for Civil Engineering Projects – Potentials and Obstacles. *ASCE Journal of Computing in Civil Engineering 25*(6), pp. 430–441. DOI: 10.1061/(ASCE)CP.1943-5487.0000135

Schrage, M. (1990). *Shared Minds: The New Technology of Collaboration*. Random House, New York.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Schwabe G., Streitz N. A., & Unland R. (2001). *CSCW-Kompendium*. Springer, Berlin, 2001. Retrieved from http://ccs.mit.edu/papers/CCSWP157.html

Searle J.R. (1976). A Classification of Illocutionary Acts. In: *Language in Society, 5*(1), pp. 1–23.

Johansen, R. (1988). *Groupware: Computer Support for Business Teams*. New York: Free Press.

Tozer, G.V. (1999). *Metadata Management for Information Control and Business Success*. Artech House computing library. Artech House. Retrieved from https://books.google.de/books?id=d9BQAAAAMAAJ

Uniclass2 (2013). *Uniclass2, the classification for the Construction Industry*. Construction Project Information Committee, UK. Retrieved from http://www.cpic.org.uk/uniclass/ (Accessed January 2015).

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Index