Abstract

We show that strong bisimilarity of CCS processes without summation and relabelling is preserved by any substitution modulo the maintenance of internal channels, if the processes bear unique input locations. By this we understand a syntactic means of preventing that substitutions, which are in general not injective, cause synchronisation in one but not in the other of two originally bisimilar processes.

1 Motivation

Substitution is one of the major features of mobile calculi such as the π-calculus, where names can be passed and substituted in the receiving process. However, it is a common fact that any of the basic bisimulation equivalences, which are not especially tailored for that purpose, fail to be congruence relations wrt. substitution of names. This can already be observed for less expressive calculi such as CCS. For instance, if we substitute \( \alpha \) for \( \beta \) in \( \alpha,\beta,0 + \beta,\alpha,0 \) and \( \alpha,0 | \beta,0 \), which are clearly strongly bisimilar, the result of the latter will be able to perform a synchronisation eating up its two components whereas the former is not. This is due to the expressibility of certain forms of concurrency by means of sequentiality and choice if communication a priori is not possible. In this way, the first process can be understood as a sequentialised implementation of the second one. When the grounds for synchronisation are provided by identifying \( \alpha \) and \( \beta \), the sequential version naturally fails to model the parallel one correctly.

As summation seems to be the main source of such nuisances, we proceed with a discussion on a subset of CCS which does not include the choice operator. For convenience, we also omit relabelling. However, we will see that even in this narrow setting, further restrictions are necessary in order to obtain preservation of strong bisimilarity by substitution.

In the following sections, we examine under which circumstances strong bisimilarity is preserved by substitution of names. This preservation depends strongly on the validity of a diamond property which we are going to introduce in the second half of section 4. Intuitively, if a process performs two different transitions, say one on \( \alpha \) and another one on \( \beta \), it will be able to enter exactly
For all $\alpha, \bar{\alpha} \in \mathit{Act}$, $\mu \in \mathit{Act}_\tau$ let $\xrightarrow{\mu} \subseteq \mathcal{P} \times \mathcal{P}$ be the smallest binary relation containing the following axiom and rules:

\[
\begin{array}{l}
\text{(pr)} \quad \mu.p \xrightarrow{\mu} p \\
\text{(m1)} \quad p \xrightarrow{\mu} p' \quad q \xrightarrow{\mu} q' \quad p|q \xrightarrow{\mu} p'|q \\
\text{(m2)} \quad q \xrightarrow{\mu} q' \quad p|q \xrightarrow{\mu} p|q' \\
\text{(cm)} \quad p \xrightarrow{\alpha} p' \quad q \xrightarrow{\beta} q' \quad p \xrightarrow{\alpha} p' q \xrightarrow{\beta} q' \quad p|q \xrightarrow{\alpha} p'|q' \\
\text{(rc)} \quad p \xrightarrow{\text{rec} \times \tau A} p' \\
\end{array}
\]

Table 1: The transition rules for $\mathcal{P}$.

the same state when performing a transition on the name used in the other transition branch. Note that nondeterministic processes may fail to fulfill this property. Take for instance $\alpha.0 + \beta.0$, which is able to perform both an $\alpha$ and a $\beta$ but not any transition afterwards.

As in the following we wish diamonds to be simulated by diamonds in the bisimulation game, further restrictions have to be made. Section 3 therefore prepares the grounds by providing a syntactical means of restricting oneself to processes fulfilling this diamond requirement. We define a subclass of processes bearing unique input locations. For such processes, no input transition, say on $\alpha$, may yield non-equivalent derivatives $p$ and $p'$.

Section 5 contains concluding remarks and proposes directions for further work on this topic.

## 2 Basic Definitions

We use $\alpha, \beta, \gamma, \ldots$ to range over the input channels $\Lambda$, and $\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \ldots$ to range over the output channels $\bar{\Lambda}$. $\mathit{Act}$ denotes the union of the disjunct sets $\Lambda$ and $\bar{\Lambda}$, $\mathit{Act}_\tau$ also comprises the silent action $\tau$. We use $\mu$ to range over $\mathit{Act}_\tau$. Channels will also be referred to as atomic actions. The name or sort $\alpha \in \Lambda$ denotes both the input channel $\alpha$ and its complement, the output channel $\bar{\alpha}$. $A, B, \ldots \subseteq \Lambda$ denote arbitrary restrictions on names.

The two mappings $\text{sort} : \mathcal{P} \rightarrow \Lambda$ and $\text{action} : \mathcal{P} \rightarrow \mathit{Act}$ determine the visible names, and the channels respectively, of processes. Both are defined in the usual syntactic way.

Let $\mathcal{P}$ be the set of CCS processes obtained in the usual way from terms of the form

\[ p ::= 0 \mid \mu.p \mid p|p \mid p \setminus A \mid \text{rec} \times p, \]

where $\mu \in \mathit{Act}_\tau$ denotes both visible and invisible action prefixes.

We use the common transition system for CCS, only excluding the laws for summation and restriction. Table 1 shows the transition rules applied in this paper.

We use the usual notion of strong bisimilarity.

**Definition 2.1 (Strong Bisimilarity)** A relation $\mathcal{R}$ on $\mathcal{P}$ is a strong bisimulation, if for all $(p, q) \in \mathcal{R}$ and $\mu \in \mathit{Act}_\tau$, the following holds:
\[ p \equiv_a p \]
\[ \mu.p\backslash A \equiv_a \mu.(p\backslash A) \quad \text{if } \mu, \nu \notin A \]
\[ (p|q)\backslash A \equiv_a p|q\backslash A \quad \text{if } \text{sort}(p) \notin A \]
\[ (p|q)\backslash A \equiv_a p\backslash A|q \quad \text{if } \text{sort}(q) \notin A \]
\[ (p|q)|r \equiv_a p|(q|r) \]

Table 2: Structural Congruence up to \( \alpha \)-Conversion.

- if \( p \xrightarrow{\mu} p' \), then \( \exists q' \text{ s.t. } q \xrightarrow{\mu} q' \) and \( (p',q') \in \mathcal{R} \).
- if \( q \xrightarrow{\mu} q' \), then \( \exists p' \text{ s.t. } p \xrightarrow{\mu} p' \) and \( (p',q') \in \mathcal{R} \).

If for processes \( p,q \in \mathcal{P} \) there exists such a bisimulation \( \mathcal{R} \) including them, we say that \( p \) and \( q \) are strongly bisimilar, written \( p \sim q \).

We apply substitution on names, strictly mapping input and output channels to such. This means that complementation commutes with substitution, i.e. for all \( \alpha \in \text{Act} \): \( \sigma(\tilde{\alpha}) = \overline{\sigma(\alpha)} \).

Just like the \( \lambda \)-operator in logics does for variables, the restriction operator \( \backslash \) acts as a binder for atomic actions which can be made internal by applying \( \backslash A \) for a subset \( A \) of \( \Lambda \) to (sub-)processes.

In the following, we will apply a structural congruence up to \( \alpha \)-conversion, \( \equiv_a \), which allows us to move around restriction sets more freely. Let \( \text{sort}(p) \) denote the set of names used in \( p \). Table 2 sums up the congruence equations.

What we need this congruence for, is to extend restriction sets from inner parts of processes to global appearance. For instance, we can write \( (p|q|\backslash r)(A \cup B) \) instead of \( (p|q|\backslash A|\backslash r)\backslash B \), as \( \alpha \)-conversion guarantees for the side condition \( \text{sort}(r) \notin A \).

3 Unique Input Location

In [BS96], Boreale and Sangiorgi give an example that replication, and thus also recursion, can cause the failure of strong bisimilarity as a congruence wrt. substitution of names. Their example hinges on the possibility of simulating the sequentialisation of certain processes using a combination of recursion and restriction. However, we will show that congruence is gained if we require the processes to yield unique input locations in the following sense.

As the transition rule for the recursion operator is not compositional in the sense that the left-hand-side of the conclusion is not made up of the left-hand-side of the premise, we cannot give a syntactical rule but one not allowing for any input locations.

**Definition 3.1** Let \( SIL : \Lambda \times \mathcal{P} \rightarrow \mathbb{N}_0 \) be the function counting for input locations \( \alpha \) at how many separate input locations within a process \( p \) \( \alpha \) occurs:

\[
SIL(\alpha, 0) = 0 \quad \text{(1)}
\]
\[
SIL(\alpha, \mu.p) = \begin{cases} \max\{1, SIL(\alpha, p)\} & \text{if } \mu \equiv \alpha \\ SIL(\alpha, p) & \text{otherwise} \end{cases} \quad \text{(2)}
\]
\[
\begin{align*}
SIL(\alpha, p | q) &= SIL(\alpha, p) + SIL(\alpha, q) \\
SIL(\alpha, p \backslash A) &= \begin{cases} SIL(\alpha, p) & \text{if } \alpha, \bar{\alpha} \notin A \\ 0 & \text{otherwise} \end{cases} \\
SIL(\alpha, \text{rec}.x.p) &= \begin{cases} 0 & \text{if } \text{actions}(p) \cap A = \emptyset \\ \infty & \text{otherwise} \end{cases}
\end{align*}
\]

**Definition 3.2** A process \( p \) bears unique input locations if for all input channels \( \alpha \) the following holds:

\[ SIL(\alpha, p) \leq 1. \]

In the sequel, we give evidence that this definition of unique input locations coincides with the intuitive notion of two input actions never being executable at different locations at the same time.

**Lemma 3.3** For all \( p, p' \in \mathcal{P}, \) where \( p \xrightarrow{\mu} p' \), and all \( \alpha \in \Lambda, \) \( SIL(\alpha, p') \leq SIL(\alpha, p). \)

**Proof:** by rule induction, where we make use of the easily deducible fact that \( SIL(\alpha, p[\text{rec}.x.p/x]) = SIL(\alpha, \text{rec}.x.p). \)

The next result may seem a bit technical, but it is as essential for the proof of lemma 3.5 as lemma 3.3.

**Lemma 3.4** For all \( p, p' \in \mathcal{P}, \alpha \in \Lambda: p \xrightarrow{\alpha} p' \) only if \( SIL(\alpha, p) > 0. \)

**Proof:** by rule induction, where \( \text{rec}.x.p \) is only capable of performing \( \alpha \) if \( SIL(\alpha, \text{rec}.x.p) = \infty. \)

The main result of this section consists of two parts. First, unique input locations are preserved by transitions. And second, they guarantee for the unicity of derivatives resulting from arbitrary input transitions.

**Lemma 3.5** If \( p \in \mathcal{P} \) is a process with unique input locations, then for every transition \( p \xrightarrow{\mu} p' \) the following holds:

(i) \( p' \) is a process with unique input locations and

(ii) if \( \mu \in \Lambda, \) then there is no \( p_1 \neq p' \) s.t. \( p \xrightarrow{\mu} p_1. \)

**Proof:**

(i) Follows directly from lemma 3.3.

(ii) By rule induction. Note that \( \text{rec}.x.p \) will never be able at all to perform \( \mu \) if it is to bear unique input locations.
Figure 1: The diamond property.

4 Congruence Results

In this section, we show that, if we restrict ourselves to processes with unique input locations as introduced above, strong bisimilarity is a congruence relation wrt. substitution of names.

**Theorem 4.1** For all \( p, q \in \mathcal{P} \) bearing unique input locations and all substitutions \( \sigma \): If \( p \sim q \) then \( p\sigma \sim q\sigma \).

For convention, we assume that domains and ranges of the substitutions are always disjunct from the internal actions of the processes which they are applied to. This is achieved by using \( \alpha \)-conversion when necessary. Besides, we assume that \( \tau \sigma \equiv \tau \) for any \( \sigma \). Obeying these two conventions, most of the proofs are straightforward inductions.

The preparatory results fall into two parts. After describing the correspondence between the transitions of processes before and after a substitution, we proceed with establishing the diamond property establishing when visible transition sequences may be subject to a synchronisation.

On the one hand, any transition of the original process can be reflected by a corresponding transition if we apply substitution:

**Lemma 4.2** For \( p, p' \in \mathcal{P} \), \( \mu \in \text{Act} \), and every substitution \( \sigma \), the following holds: If \( p \xrightarrow{\mu} p' \) then \( p\sigma \xrightarrow{\mu\sigma} p'\sigma \).

**Proof:** by rule induction. \( \square \)

On the other hand, also the reverse direction holds, where we only have to take into account communication that may not exist in the original process but may well be possible after a non-injective substitution.

**Lemma 4.3** For every \( p, p' \in \mathcal{P} \) and substitution \( \sigma \), the following holds:

(i) Whenever \( p\sigma \xrightarrow{\beta} p' \), there exist \( \alpha, p_1 \) s.t. \( p \xrightarrow{\alpha} p_1 \), \( \beta \equiv \alpha\sigma \) and \( p' \equiv p_1\sigma \).

(ii) Whenever \( p\sigma \xrightarrow{\tau} p' \),

1. either there exists \( p_1 \) s.t. \( p \xrightarrow{\tau} p_1 \) and \( p' \equiv p_1\sigma \).
2. or there exist \( \alpha, \beta, p_1, p_2, p_3 \) s.t. \( p \xrightarrow{\alpha} p_1 \xrightarrow{\beta} p_3, \quad p \xrightarrow{\beta} p_2 \xrightarrow{\alpha} p_3, \quad \alpha \sigma \equiv \beta \sigma \) and \( p' \equiv p_3 \sigma \).

**Proof:** by rule induction. \( \square \)

The next two results represent the crucial part of our proof. They guarantee that for bisimilar \( p \) and \( q \) and some substitution \( \sigma \), if \( p \sigma \) can perform a silent step which is the result of two non-matching visible transitions of \( p \) (and hence of \( q \)), this also leads to a corresponding synchronisation of \( q \sigma \).

**Lemma 4.4 (Diamond Property)** For all \( p, p_1, p_2, \in \mathcal{P} \) and \( \alpha, \beta \in \text{Act} \) where \( \alpha \not\equiv \beta \), the following holds:

(i) If \( p \xrightarrow{\alpha} p_1 \) and \( p \xrightarrow{\beta} p_2 \), then there exists \( p_3 \) s.t. \( p_1 \xrightarrow{\beta} p_3 \) and \( p_2 \xrightarrow{\alpha} p_3 \).

(ii) If \( \alpha \equiv \beta \) then \( p \xrightarrow{\tau} p_3 \).

**Proof:** by rule induction. \( \square \)

Figure 1 (a) illustrates why the term *diamond* was chosen for this property.

**Remark:** New diamonds arise when both components of a process \( p \mid p \) are able to perform (different) visible transitions, as shown in figure 1 (b).

The diamond lemma is straightforward to establish if we require *unique input locations*.

**Lemma 4.5 (Diamond Lemma)** For all \( p, q, \in \mathcal{P} \), \( \alpha \in \Lambda \) and \( \tilde{\beta} \in \tilde{\Lambda} \), the following holds:

If \( p \sim q \) and \( p \) performs a diamond on \( \alpha \) and \( \tilde{\beta} \) yielding \( p_3 \), also \( q \) can perform a diamond on \( \alpha \) and \( \tilde{\beta} \) yielding some \( q_3 \) which is bisimilar to \( p_3 \).

Yet, we do not only prove the existence of such a diamond, but show that the diamonds for \( p \) and \( q \) are bisimilar at every step.

**Proof:** If there is a diamond for \( p \), then certainly there exists a diamond for \( q \), as \( p \xrightarrow{\alpha} p_1 \) has to be simulated by some \( q \xrightarrow{\alpha} q_1 \) and for \( p \xrightarrow{\beta} p_2 \) there has to be a transition \( q \xrightarrow{\tilde{\beta}} q_2 \). By lemma 4.4 there exists some \( q_3 \) completing the diamond.

As both \( p \) and \( q \) bear unique input locations, by lemma 3.5 also \( p_1, p_2, q_1 \) and \( q_2 \) do. Hence, the only possibility for \( q \) to perform an \( \alpha \) yields \( q_1 \), which therefore has to be bisimilar with \( p_1 \). Let \( q_2 \) be a \( \tilde{\beta} \)-derivative of \( q \) s.t. \( p_2 \sim q_2 \). As \( q_2 \) bears unique input locations (see above), the only possibility for it to perform \( \alpha \) is to enter \( q_3 \) which, again, has to be bisimilar with \( p_3 \). Hence, the only diamond on \( \alpha \) and \( \tilde{\beta} \) existing yields \( q_3 \), i.e. by lemma 4.4 also \( q_1 \xrightarrow{\tilde{\beta}} q_3 \).

\( \square \)

With this material, we are now able to prove the main result of this work.

**Theorem 4.1** For all \( p, q \in \mathcal{P} \) bearing unique input locations and all substitutions \( \sigma \): If \( p \sim q \) then \( p \sigma \sim q \sigma \).
Proof: It suffices to show that, for any substitution $\sigma$, the relation $R^\sim_\sigma \overset{\text{def}}{=} \{ (p\sigma, q\sigma) \mid p \sim q \}$ is a strong bisimulation.

Let $p \sim q$. We presume a transition $p\sigma \xrightarrow{\mu} p'$ which has to be simulated by $q\sigma \xrightarrow{\mu} q'$ in such a way that $(p', q') \in R^\sim_\sigma$. We proceed by a case study on $\mu$:

- **Visible transitions:** $\mu \equiv \beta \in \text{Act}$.

  If $p\sigma \xrightarrow{\beta} p'$ then, by lemma 4.3, there exist $\alpha$ and $p_1$ s.t. $p \xrightarrow{\alpha} p_1$, $\beta \equiv \alpha\sigma$ and $p' \equiv p_1\sigma$. As $p \sim q$, by definition 2.1 there must exist some $q_1$ s.t. $q \xrightarrow{\alpha} q_1$ and $p_1 \sim q_1$. By lemma 4.2, $q\sigma \xrightarrow{\beta} q_1\sigma$. Hence, $(p', q') \in R^\sim_\sigma$, where $p' \equiv p_1\sigma$ and $q' \equiv q_1\sigma$.

- **Invisible transitions:** $\mu \equiv \tau$.

  Now let $p\sigma \xrightarrow{\tau} p'$. By lemma 4.3,

  - either $\exists p_1$ s.t. $p \xrightarrow{\tau} p_1$ and $p' \equiv p_1\sigma$, which is similar to the case of visible transitions.

  - or $\exists \alpha, \beta, p_1, p_2, p_3$ s.t. $p \xrightarrow{\alpha} p_1 \xrightarrow{\beta} p_3$, $p \xrightarrow{\beta} p_2 \xrightarrow{\alpha} p_3$, $\alpha\sigma \equiv \beta\sigma$ and $p' \equiv p_3\sigma$. As $p$ and $q$ are bisimilar, $q$ must be able to imitate the $\alpha$- and $\beta$-transitions which $p$ has performed. As $\alpha \not\equiv \beta$, we know from lemma 4.5 that there exists an $\alpha$-$\beta$-diamond for $q$, its tip $\bar{q}_3$ being bisimilar with $\bar{p}_3$. Clearly, $(p_3\sigma, q_3\sigma) \in R^\sim_\sigma$. Furthermore, this diamond extends to $q\sigma \xrightarrow{\alpha\sigma} q_1\sigma \xrightarrow{\beta\sigma} q_3\sigma$ and $q\sigma \xrightarrow{\beta\sigma} q_2\sigma \xrightarrow{\alpha\sigma} q_3\sigma$.

  As $\alpha\sigma \equiv \beta\sigma$, we know from the second part of lemma 4.4 that $q\sigma \xrightarrow{\tau} q_3\sigma$.

- A symmetric argument holds for the transitions initiated by $q$. \qed

5 Discussion

In this work, we have shown that for CCS-processes obeying certain restrictions, strong bisimilarity is a congruence wrt. substitution of names. As the exclusion of the summation operator does not yet fully suffice, we proposed unique input locations as a way of further reducing the set of processes. However, we conjecture that for finite processes, i.e. if we also omit recursion, this artificial construct would not be necessary.

Another approach would consist of applying equivalence relations which fulfill the diamond property by themselves, as do local cause or global cause equivalence for large subsets of CCS.

For weak bisimulations the diamond property does not hold. Although the process $\alpha, \beta, 0 | \delta, 0$, for instance, is capable of performing a weak transition both on $\alpha$ and $\beta$, the diamond can never be completed. Therefore, one would have to find a new notion of diamond property relying on the use of unique input locations.

As substitutions play a major role in mobile calculi, e.g. the $\pi$-calculus, it would...
also be interesting to study how the results obtained here extend thereupon.

Acknowledgement: I would like to thank Davide Sangiorgi for proposing this problem and for giving helpful comments.

References


SFB 342: Methoden und Werkzeuge für die Nutzung paralleler
Rechnerarchitekturen

bisher erschienen:

Reihe A

342/2/90 A Reinhard Fößmeier: Die Rolle der Lastverteilung bei der numerischen Parallelprogrammierung, Februar 1990
342/3/90 A Klaus-Jörn Lange, Peter Rossmanith: Two Results on Unambiguous Circuits, Februar 1990
342/5/90 A Reinhold Letz, Johann Schumann, Stephan Bayerl, Wolfgang Bibel: SETHEO: A High-Performance Theorem Prover
342/6/90 A Johann Schumann, Reinhold Letz: PARTHEO: A High Performance Parallel Theorem Prover
342/7/90 A Johann Schumann, Norbert Trapp, Martin van der Koelen: SETHEO/PARTHEO Users Manual
342/10/90 A Walter Vogler: Bisimulation and Action Refinement
342/11/90 A Jörg Desel, Javier Esparza: Reachability in Reversible Free-Choice Systems
342/12/90 A Rob van Glabbeck, Ursula Goltz: Equivalences and Refinement
342/13/90 A Rob van Glabbeck: The Linear Time - Branching Time Spectrum
342/14/90 A Johannes Bauer, Thomas Bemmerl, Thomas Treml: Leistungsanalyse von verteilten Beobachtungs- und Bewertungswerkzeugen
342/15/90 A Peter Rossmanith: The Owner Concept for PRAMs
342/16/90 A G. Böckle, S. Trosch: A Simulator for VLIW-Architectures
342/17/90 A P. Slavkovsky, U. Rüde: Schnellere Berechnung klassischer Matrix-Multiplikationen
342/18/90 A Christoph Zenger: SPARSE GRIDS
Reihe A

342/19/90 A Michael Griebel, Michael Schneider, Christoph Zenger: A combination technique for the solution of sparse grid problems

342/20/90 A Michael Griebel: A Parallelizable and Vectorizable Multi-Level-Algorithm on Sparse Grids

342/21/90 A V. Diekert, E. Ochmanski, K. Reinhardt: On confluent semicommutations-decidability and complexity results

342/22/90 A Manfred Broy, Claus Dendorfer: Functional Modelling of Operating System Structures by Timed Higher Order Stream Processing Functions

342/23/90 A Rob van Glabbeek, Ursula Goltz: A Deadlock-sensitive Congruence for Action Refinement

342/24/90 A Manfred Broy: On the Design and Verification of a Simple Distributed Spanning Tree Algorithm


342/27/90 A Wolfgang Ertel: Random Competition: A Simple, but Efficient Method for Parallelizing Inference Systems

342/28/90 A Rob van Glabbeek, Frits Vaandrager: Modular Specification of Process Algebras

342/29/90 A Rob van Glabbeek, Peter Weijland: Branching Time and Abstraction in Bisimulation Semantics

342/30/90 A Michael Griebel: Parallel Multigrid Methods on Sparse Grids

342/31/90 A Rolf Niedermeier, Peter Rossmanith: Unambiguous Simulations of Auxiliary Pushdown Automata and Circuits

342/32/90 A Inga Niepel, Peter Rossmanith: Uniform Circuits and Exclusive Read PRAMs

342/33/90 A Dr. Hermann Hellwagner: A Survey of Virtually Shared Memory Schemes

342/1/91 A Walter Vogler: Is Partial Order Semantics Necessary for Action Refinement?

342/2/91 A Manfred Broy, Frank Dederichs, Claus Dendorfer, Rainer Weber: Characterizing the Behaviour of Reactive Systems by Trace Sets

342/3/91 A Ulrich Furbach, Christian Suttner, Bertram Fronhöfer: Massively Parallel Inference Systems

342/4/91 A Rudolf Bayer: Non-deterministic Computing, Transactions and Recursive Atomicity

342/5/91 A Robert Gold: Dataflow semantics for Petri nets

342/6/91 A A. Heise; C. Dimitrovic: Transformation und Komposition von P/T-Netzen unter Erhaltung wesentlicher Eigenschaften

342/7/91 A Walter Vogler: Asynchronous Communication of Petri Nets and the Refinement of Transitions

342/8/91 A Walter Vogler: Generalized OM-Bisimulation

342/9/91 A Christoph Zenger, Klaus Hallatschek: Fouriertransformation auf dünnen Gitternm mit hierarchischen Basen
Reihe A

342/10/91 A Erwin Loibl, Hans Obermaier, Markus Pawlowski: Towards Parallelism in a Relational Database System

342/11/91 A Michael Werner: Implementierung von Algorithmen zur Kompaktifizierung von Programmen für VLIW-Architekturen

342/12/91 A Reiner Müller: Implementierung von Algorithmen zur Optimierung von Schleifen mit Hilfe von Software-Pipelining Techniken

342/13/91 A Sally Baker, Hans-Jörg Beier, Thomas Benmerl, Arndt Bode, Hubert Erdl, Udo Graf, Olav Hansen, Josef Haumerdinger, Paul Hofstetter, Rainer Knödlseder, Jaroslav Kremenek, Siegfried Langenbuch, Robert Lindhof, Thomas Ludwig, Peter Luksch, Roy Mihner, Bernhard Ries, Thomas Treml: TOPSYS - Tools for Parallel Systems (Artikelsammlung); 2., erweiterte Auflage

342/14/91 A Michael Griebel: The combination technique for the sparse grid solution of PDE’s on multiprocessor machines

342/15/91 A Thomas F. Gritzner, Manfred Broy: A Link Between Process Algebras and Abstract Relation Algebras?

342/16/91 A Thomas Benmerl, Arndt Bode, Peter Braun, Olav Hansen, Thomas Treml, Roland Wismüller: The Design and Implementation of TOPSYS

342/17/91 A Ulrich Furbach: Answers for disjunctive logic programs

342/18/91 A Ulrich Furbach: Splitting as a source of parallelism in disjunctive logic programs

342/19/91 A Gerhard W. Zumbusch: Adaptive parallele Multilevel-Methoden zur Lösung elliptischer Randwertprobleme

342/20/91 A M. Jobmann, J. Schumann: Modelling and Performance Analysis of a Parallel Theorem Prover


342/22/91 A Wolfgang Ertel, Theodor Gemenis, Johann M. Ph. Schumann, Christian B. Suttner, Rainer Weber, Zongyan Qiu: Formalisms and Languages for Specifying Parallel Inference Systems

342/23/91 A Astrid Kiehn: Local and Global Causes

342/24/91 A Johann M. Ph. Schumann: Parallelization of Inference Systems by using an Abstract Machine

342/25/91 A Eike Jessen: Speedup Analysis by Hierarchical Load Decomposition


342/27/91 A Thomas Schneekenburger, Andreas Weininger, Michael Friedrich: Introduction to the Parallel and Distributed Programming Language ParMod-C

342/28/91 A Claus Dendorfer: Funktionale Modellierung eines Postsystems

342/29/91 A Michael Griebel: Multilevel algorithms considered as iterative methods on indefinite systems

342/30/91 A W. Reisig: Parallel Composition of Liveness

342/31/91 A Thomas Benmerl, Christian Kasperbauer, Martin Mairandres, Bernhard Ries: Programming Tools for Distributed Multiprocessor Computing Environments
Reihe A

342/32/91 A Frank Leßke: On constructive specifications of abstract data types using temporal logic
342/1/92 A L. Kanal, C.B. Suttner (Editors): Informal Proceedings of the Workshop on Parallel Processing for AI
342/2/92 A Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: The Design of Distributed Systems - An Introduction to FOCUS
342/3/92 A Manfred Broy, Frank Dederichs, Claus Dendorfer, Max Fuchs, Thomas F. Gritzner, Rainer Weber: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems
342/5/92 A Michael Friedrich: Sprachmittel und Werkzeuge zur Unterstützung paralleler und verteilter Programmierung
342/6/92 A Thomas F. Gritzner: The Action Graph Model as a Link between Abstract Relation Algebras and Process-Algebraic Specifications
342/7/92 A Sergei Gorlatch: Parallel Program Development for a Recursive Numerical Algorithm: a Case Study
342/8/92 A Henning Spruth, Georg Sigl, Frank Johannes: Parallel Algorithms for Slicing Based Final Placement
342/9/92 A Herbert Bauer, Christian Sporrer, Thomas Krodel: On Distributed Logic Simulation Using Time Warp
342/10/92 A H. Bungartz, M. Griebel, U. Rüde: Extrapolation, Combination and Sparse Grid Techniques for Elliptic Boundary Value Problems
342/11/92 A M. Griebel, W. Huber, U. Rüde, T. Störtkuhl: The Combination Technique for Parallel Sparse-Grid-Preconditioning and -Solution of PDEs on Multiprocessor Machines and Workstation Networks
342/12/92 A Rolf Niedermeier, Peter Rossmanith: Optimal Parallel Algorithms for Computing Recursively Defined Functions
342/13/92 A Rainer Weber: Eine Methodik für die formale Anforderungsspezifikation verteilter Systeme
342/14/92 A Michael Griebel: Grid- and point-oriented multilevel algorithms
342/15/92 A M. Griebel, C. Zenger, S. Zimmer: Improved multilevel algorithms for full and sparse grid problems
342/16/92 A J. Desel, D. Gomm, E. Kindler, B. Paech, R. Walter: Bausteine eines kompositionalen Beweiskalküls für netzmodellierte Systeme
342/17/92 A Frank Dederichs: Transformation verteilter Systeme: Von applikativen zu prozeduralen Darstellungen
342/18/92 A Andreas Listl, Markus Pawlowski: Parallel Cache Management of a RDBMS
342/19/92 A Erwin Loibl, Markus Pawlowski, Christian Roth: PART: A Parallel Relational Toolbox as Basis for the Optimization and Interpretation of Parallel Queries
<table>
<thead>
<tr>
<th>Artikelnummer</th>
<th>Autor</th>
<th>Titel</th>
</tr>
</thead>
<tbody>
<tr>
<td>342/20/92</td>
<td>Jörg Desel, Wolfgang Reisig</td>
<td>The Synthesis Problem of Petri Nets</td>
</tr>
<tr>
<td>342/21/92</td>
<td>Robert Balder, Christoph Zenger</td>
<td>The d-dimensional Helmholtz equation on sparse Grids</td>
</tr>
<tr>
<td>342/22/92</td>
<td>Ilko Michler</td>
<td>Neuronale Netzwerk-Paradigmen zum Erlernen von Heuristiken</td>
</tr>
<tr>
<td>342/23/92</td>
<td>Wolfgang Reisig</td>
<td>Elements of a Temporal Logic. Coping with Concurrency</td>
</tr>
<tr>
<td>342/24/92</td>
<td>T. Störtkuhl, Chr. Zenger, S. Zimmer</td>
<td>An asymptotic solution for the singularity at the angular point of the lid driven cavity</td>
</tr>
<tr>
<td>342/25/92</td>
<td>Ekkart Kindler</td>
<td>Invariants, Compositionality and Substitution</td>
</tr>
<tr>
<td>342/26/92</td>
<td>Thomas Bonk, Ulrich Rüde</td>
<td>Performance Analysis and Optimization of Numerically Intensive Programs</td>
</tr>
<tr>
<td>342/1/93</td>
<td>M. Griebel, V. Thurner</td>
<td>The Efficient Solution of Fluid Dynamics Problems by the Combination Technique</td>
</tr>
<tr>
<td>342/2/93</td>
<td>Ketil Stølen, Frank Dederichs, Rainer Weber</td>
<td>Assumption / Commitment Rules for Networks of Asynchronously Communicating Agents</td>
</tr>
<tr>
<td>342/3/93</td>
<td>Thomas Schneckenburger</td>
<td>A Definition of Efficiency of Parallel Programs in Multi-Tasking Environments</td>
</tr>
<tr>
<td>342/4/93</td>
<td>Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger</td>
<td>A Proof of Convergence for the Combination Technique for the Laplace Equation Using Tools of Symbolic Computation</td>
</tr>
<tr>
<td>342/5/93</td>
<td>Manfred Kunde, Rolf Niedermeier, Peter Rossmanith</td>
<td>Faster Sorting and Rounting on Grids with Diagonals</td>
</tr>
<tr>
<td>342/6/93</td>
<td>Michael Griebel, Peter Oswald</td>
<td>Remarks on the Abstract Theory of Additive and Multiplicative Schwarz Algorithms</td>
</tr>
<tr>
<td>342/7/93</td>
<td>Christian Sporrer, Herbert Bauer</td>
<td>Corolla Partitioning for Distributed Logic Simulation of VLSI Circuits</td>
</tr>
<tr>
<td>342/8/93</td>
<td>Herbert Bauer, Christian Sporrer</td>
<td>Reducing Rollback Overhead in Time-Warp Based Distributed Simulation with Optimized Incremental State Saving</td>
</tr>
<tr>
<td>342/9/93</td>
<td>Peter Slavkovsky</td>
<td>The Visibility Problem for Single-Valued Surface (z = f(x,y)): The Analysis and the Parallelization of Algorithms</td>
</tr>
<tr>
<td>342/10/93</td>
<td>Ulrich Rüde</td>
<td>Multilevel, Extrapolation, and Sparse Grid Methods</td>
</tr>
<tr>
<td>342/11/93</td>
<td>Hans Regler, Ulrich Rüde</td>
<td>Layout Optimization with Algebraic Multi-grid Methods</td>
</tr>
<tr>
<td>342/12/93</td>
<td>Dieter Barnard, Angelika Mader</td>
<td>Model Checking for the Modal Mu-Calculus using Gauß Elimination</td>
</tr>
<tr>
<td>342/13/93</td>
<td>Christoph Pflaum, Ulrich Rüde</td>
<td>Gauß' Adaptive Relaxation for the Multilevel Solution of Partial Differential Equations on Sparse Grids</td>
</tr>
<tr>
<td>342/14/93</td>
<td>Christoph Pflaum</td>
<td>Convergence of the Combination Technique for the Finite Element Solution of Poisson's Equation</td>
</tr>
<tr>
<td>342/15/93</td>
<td>Michael Luby, Wolfgang Ertel</td>
<td>Optimal Parallelization of Las Vegas Algorithms</td>
</tr>
<tr>
<td>342/16/93</td>
<td>Hans-Joachim Bungartz, Michael Griebel, Dierk Röschke, Christoph Zenger</td>
<td>Pointwise Convergence of the Combination Technique for Laplace's Equation</td>
</tr>
</tbody>
</table>
Reihe A

342/17/93 A Georg Stellner, Matthias Schumann, Stefan Lamberts, Thomas Ludwig, Arndt Bode, Martin Kiehl und Rainer Mehlhorn: Developing Multicomputer Applications on Networks of Workstations Using NXLib

342/18/93 A Max Fuchs, Ketil Stølen: Development of a Distributed Min/Max Component

342/19/93 A Johann K. Obermaier: Recovery and Transaction Management in Write-optimized Database Systems

342/20/93 A Sergej Gorlatch: Deriving Efficient Parallel Programs by Systematizing Coarsening Specification Parallelism

342/01/94 A Reiner Hüttl, Michael Schneider: Parallel Adaptive Numerical Simulation

342/02/94 A Henning Spruth, Frank Johannes: Parallel Routing of VLSI Circuits Based on Net Independence

342/03/94 A Henning Spruth, Frank Johannes, Kurt Antreich: PHIroute: A Parallel Hierarchical Sea-of-Gates Router

342/04/94 A Martin Kiehl, Rainer Mehlhorn, Matthias Schumann: Parallel Multiple Shooting for Optimal Control Problems Under NX/2

342/05/94 A Christian Suttner, Christoph Goller, Peter Krauss, Klaus-Jörn Lange, Ludwig Thomas, Thomas Schneekenburger: Heuristic Optimization of Parallel Computations

342/06/94 A Andreas Listl: Using Subpages for Cache Coherency Control in Parallel Database Systems

342/07/94 A Manfred Broy, Ketil Stølen: Specification and Refinement of Finite Dataflow Networks - a Relational Approach

342/08/94 A Katharina Spies: Funktionale Spezifikation eines Kommunikationsprotokolls

342/09/94 A Peter A. Krauss: Applying a New Search Space Partitioning Method to Parallel Test Generation for Sequential Circuits

342/10/94 A Manfred Broy: A Functional Rephrasing of the Assumption/Commitment Specification Style

342/11/94 A Eckhardt Holz, Ketil Stølen: An Attempt to Embed a Restricted Version of SDL as a Target Language in Focus


342/13/94 A Manfred Broy, Max Fuchs, Thomas F. Gritzner, Bernhard Schütz, Katharina Spies, Ketil Stølen: Summary of Case Studies in FOCUS - a Design Method for Distributed Systems

342/14/94 A Maximilian Fuchs: Technologienabhängigkeit von Spezifikationen digitaler Hardware

342/15/94 A M. Griebel, P. Oswald: Tensor Product Type Subspace Splittings And Multilevel Iterative Methods For Anisotropic Problems

342/16/94 A Gheorghe Ștefănescu: Algebra of Flownomials

342/17/94 A Ketil Stølen: A Refinement Relation Supporting the Transition from Unbounded to Bounded Communication Buffers

342/18/94 A Michael Griebel, Tilman Neuhoefner: A Domain-Oriented Multilevel Algorithm-Implementation and Parallelization
<table>
<thead>
<tr>
<th>Year</th>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>Michael Griebel, Walter Huber</td>
<td>Turbulence Simulation on Sparse Grids Using the Combination Method</td>
</tr>
<tr>
<td>1994</td>
<td>Johann Schumann</td>
<td>Using the Theorem Prover SETHEO for verifying the development of a Communication Protocol in FOCUS - A Case Study -</td>
</tr>
<tr>
<td>1995</td>
<td>Hans-Joachim Bungartz</td>
<td>Higher Order Finite Elements on Sparse Grids</td>
</tr>
<tr>
<td>1995</td>
<td>Tao Zhang, Seonglim Kang, Lester R. Lipsky</td>
<td>The Performance of Parallel Computers: Order Statistics and Amdahl's Law</td>
</tr>
<tr>
<td>1995</td>
<td>Lester R. Lipsky, Apple van de Liefvoort</td>
<td>Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space</td>
</tr>
<tr>
<td>1995</td>
<td>Sascha Hilgenfeldt, Robert Balder, Christoph Zenger</td>
<td>Sparse Grids: Applications to Multi-dimensional Schrödinger Problems</td>
</tr>
<tr>
<td>1995</td>
<td>Maximilian Fuchs</td>
<td>Formal Design of a Model-N Counter</td>
</tr>
<tr>
<td>1995</td>
<td>Hans-Joachim Bungartz, Stefan Schulte</td>
<td>Coupled Problems in Microsystems Technology</td>
</tr>
<tr>
<td>1995</td>
<td>Alexander Pfallinger</td>
<td>Parallel Communication on Workstation Networks with Complex Topologies</td>
</tr>
<tr>
<td>1995</td>
<td>Ketil Stølen</td>
<td>Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness</td>
</tr>
<tr>
<td>1995</td>
<td>Ketil Stølen, Max Fuchs</td>
<td>A Formal Method for Hardware/Software Co-Design</td>
</tr>
<tr>
<td>1995</td>
<td>Thomas Schneekenburger</td>
<td>The ALDY Load Distribution System</td>
</tr>
<tr>
<td>1995</td>
<td>Javier Esparza, Stefan Römer, Walter Vogler</td>
<td>An Improvement of McMillan's Unfolding Algorithm</td>
</tr>
<tr>
<td>1995</td>
<td>Stephan Melzer, Javier Esparza</td>
<td>Checking System Properties via Integer Programming</td>
</tr>
<tr>
<td>1995</td>
<td>Radu Grosu, Ketil Stølen</td>
<td>A Denotational Model for Mobile Point-to-Point Dataflow Networks</td>
</tr>
<tr>
<td>1995</td>
<td>Andrei Kovalyov, Javier Esparza</td>
<td>A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs</td>
</tr>
<tr>
<td>1995</td>
<td>Bernhard Schätz, Katharina Spies</td>
<td>Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik</td>
</tr>
<tr>
<td>1995</td>
<td>Georg Stellner</td>
<td>Using CoCheck on a Network of Workstations</td>
</tr>
<tr>
<td>1995</td>
<td>Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller</td>
<td>Workshop on PVM, MPI, Tools and Applications</td>
</tr>
<tr>
<td>1995</td>
<td>Thomas Schneekenburger</td>
<td>Integration of Load Distribution into ParMod-C</td>
</tr>
<tr>
<td>1995</td>
<td>Ketil Stølen</td>
<td>Refinement Principles Supporting the Transition from Asynchronous to Synchronous Communication</td>
</tr>
<tr>
<td>1995</td>
<td>Andreas Listl, Giannis Bozas</td>
<td>Performance Gains Using Subpages for Cache Coherency Control</td>
</tr>
<tr>
<td>1995</td>
<td>Volker Heum, Ernst W. Mayr</td>
<td>Embedding Graphs with Bounded Treewidth into Optimal Hypercubes</td>
</tr>
<tr>
<td>Volume</td>
<td>Title</td>
<td>Authors</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>23/95</td>
<td>Deciding Finiteness of Petri Nets up to Bisimulation</td>
<td>Petr Jančar, Javier Esparza</td>
</tr>
<tr>
<td>24/95</td>
<td>Implicit Extrapolation Methods for Variable Coefficient Problems</td>
<td>M. Jung, U. Rüde</td>
</tr>
<tr>
<td>01/96</td>
<td>Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries</td>
<td>Michael Griebel, Tilman Neunhoeffer, Hans Regler</td>
</tr>
<tr>
<td>02/96</td>
<td>Additive Multilevel-Preconditioners based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic-Multigrid Coarsening for Second Order Elliptic PDEs</td>
<td>Thomas Grauschopf, Michael Griebel, Hans Regler</td>
</tr>
<tr>
<td>03/96</td>
<td>Optimal Dynamic Edge-Disjoint Embeddings of Complete Binary Trees into Hypercubes</td>
<td>Volker Heun, Ernst W. Mayer</td>
</tr>
<tr>
<td>04/96</td>
<td>Efficient Computation of Sparse Approximate Inverses</td>
<td>Thomas Huckle</td>
</tr>
<tr>
<td>05/96</td>
<td>OMIS — On-line Monitoring Interface Specification</td>
<td>Thomas Ludwig, Roland Wismüller, Vaidy Sunderam, Arndt Bode</td>
</tr>
<tr>
<td>06/96</td>
<td>A Compositional Partial Order Semantics for Petri Net Components</td>
<td>Eckart Kindler</td>
</tr>
<tr>
<td>07/96</td>
<td>Some Results on Basic Parallel Processes</td>
<td>Richard Mayer</td>
</tr>
<tr>
<td>08/96</td>
<td>INSEL Syntax-Bericht</td>
<td>Ralph Radermacher, Frank Weimer</td>
</tr>
<tr>
<td>10/96</td>
<td>PFS-Lib — A File System for Parallel Programming Environments</td>
<td>Stefan Lamberts, Thomas Ludwig, Christian Röder, Arndt Bode</td>
</tr>
<tr>
<td>11/96</td>
<td>The Algebra of Stream Processing Functions</td>
<td>Manfred Broy, Gheorghe Ștefanescu</td>
</tr>
<tr>
<td>12/96</td>
<td>Reachability in Live and Safe Free-Choice Petri Nets is NP-complete</td>
<td>Javier Esparza</td>
</tr>
<tr>
<td>13/96</td>
<td>A Denotational Model for Mobile Many-to-Many Data-flow Networks</td>
<td>Radu Grosu, Ketil Stolen</td>
</tr>
<tr>
<td>14/96</td>
<td>On Transforming a Sequential SQL-DBMS into a Parallel One: First Results and Experiences of the MIDAS Project</td>
<td>Giannis Bozas, Michael Jaedicke, Andreas Listl, Bernhard Mitschang, Angelika Reiser, Stephan Zimmermann</td>
</tr>
<tr>
<td>15/96</td>
<td>A Tableau System for Model Checking Petri Nets with a Fragment of the Linear Time (\mu)-Calculus</td>
<td>Richard Mayer</td>
</tr>
<tr>
<td>16/96</td>
<td>Anleitung zur Spezifikation von mobilen, dynamischen Focus-Netzen</td>
<td>Ursula Hinkel, Katharina Spies</td>
</tr>
<tr>
<td>17/96</td>
<td>Model Checking PA-Processes</td>
<td>Richard Mayer</td>
</tr>
<tr>
<td>18/96</td>
<td>Putting your Model Checker on Diet: Verification on Local States</td>
<td>Michaela Huhn, Peter Niebert, Frank Wallner</td>
</tr>
<tr>
<td>01/97</td>
<td>Evaluierung der Leistungsfähigkeit eines ATM-Netzes mit parallelen Programmierbibliotheken</td>
<td>Tobias M&quot;uller, Stefan Lamberts, Ursula Maier, Georg Stellner</td>
</tr>
<tr>
<td>02/97</td>
<td>Sparse Grids: Recent Developments for Elliptic Partial Differential Equations</td>
<td>Hans-Joachim Bungartz and Thomas Dornseifer</td>
</tr>
</tbody>
</table>
Reihe A

342/03/97 A Bernhard Mitschang: Technologie f"ur Parallele Datenbanken - Bericht
zum Workshop
342/04/97 A nicht erschienen
342/05/97 A Hans-Joachim Bungartz, Ralf Ebner, Stefan Schulte: Hierarchische
Basen zur effizienten Kopplung substrukturierter Probleme der Strukturmechanik
342/06/97 A Hans-Joachim Bungartz, Anton Frank, Florian Meier, Tilman Neunhoeffer, Stefan Schulte: Fluid Structure Interaction: 3D Numerical Simulation and Visualization of a Micropump
342/07/97 A Javier Esparza, Stephan Melzer: Model Checking LTL using Constraint Programming
342/08/97 A Niels Reimer: Untersuchung von Strategien für verteiltes Last- und Ressourcenmanagement
342/09/97 A Markus Pizka: Design and Implementation of the GNU INSEL-Compiler
342/10/97 A Manfred Broy, Franz Regensburger, Bernhard Schätz, Katharina Spies: The Steamboiler Specification - A Case Study in Focus
342/11/97 A Christine Röckl: How to Make Substitution Preserve Strong Bisimilarity
SFB 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

Reihe B

342/1/90 B Wolfgang Reisig: Petri Nets and Algebraic Specifications
342/2/90 B Jörg Desel: On Abstraction of Nets
342/3/90 B Jörg Desel: Reduction and Design of Well-behaved Free-choice Systems
342/4/90 B Franz Abstreiter, Michael Friedrich, Hans-Jürgen Plewan: Das Werkzeug runtime zur Beobachtung verteilter und paralleler Programme
342/1/91 B Barbara Paecht: Concurrency as a Modality
342/2/91 B Birgit Kandler, Markus Pawlowski: SAM: Eine Sortier-Toolbox-Anwenderbeschreibung
342/3/91 B Erwin Loibl, Hans Obermaier, Markus Pawlowski: 2. Workshop über Parallelisierung von Datenbanksystemen
342/4/91 B Werner Pohlmann: A Limitation of Distributed Simulation Methods
342/5/91 B Dominik Gomm, Ekkart Kindler: A Weakly Coherent Virtually Shared Memory Scheme: Formal Specification and Analysis
342/6/91 B Dominik Gomm, Ekkart Kindler: Causality Based Specification and Correctness Proof of a Virtually Shared Memory Scheme
342/7/91 B W. Reisig: Concurrent Temporal Logic
342/1/92 B Malte Grosse, Christian B. Suttner: A Parallel Algorithm for Set-of-Support
            Christian B. Suttner: Parallel Computation of Multiple Sets-of-Support
342/2/92 B Arndt Bode, Hartmut Wedekind: Parallelrechner: Theorie, Hardware, Software, Anwendungen
342/1/93 B Max Fuchs: Funktionale Spezifikation einer Geschwindigkeitsregelung
342/2/93 B Ekkart Kindler: Sicherheits- und Lebendigkeitseigenschaften: Ein Literaturüberblick
342/1/94 B Andreas Listl; Thomas Schnekenburger; Michael Friedrich: Zum Entwurf eines Prototypen für MIDAS