TECHNISCHE UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK

Sonderforschungsbereich 342:
Methoden und Werkzeuge für die Nutzung
paralleler Rechnerarchitekturen

Summary of Case Studies in Focus - Part II

Manfred Broy Max Breitling Bernhard Schätz Katharina Spies

TUM-I9740
SFB-Bericht Nr. 342/24/97 A
September 97

TUM–INFO–09-I9740-200/1.–FI

Anforderungen an: Prof. Dr. A. Bode
Sprecher SFB 342
Institut für Informatik
Technische Universität München
D-80290 München, Germany

Druck: Fakultät für Informatik der
Technischen Universität München

Summary of Case Studies in FOCUS[*]
- Part II -

Manfred Broy
Max Breitling
Bernhard Schätz
Katharina Spies

Institut für Informatik
Technische Universität München
D-80290 München

http://www4.informatik.tu-muenchen.de/

September 5, 1997

Abstract

This paper is the continuation of a survey of case studies that are have been carried out using the methodology FOCUS. After a short introduction to FOCUS, the essential aspects of the case studies are shortly described, and the relevant publications are referenced for further reading.

[*]This work is supported by the Sonderforschungsbereich 342 "Werkzeuge und Methoden für die Nutzung paralleler Rechnerarchitekturen", funded by DFG.

Contents

1 Focus - A Design Methodology for Distributed Systems

Following the Focus methodology, a system development process goes through several levels of abstraction starting at an abstract requirement specification and going down to a concrete implementation description. During the *requirement phase* a first and possibly very abstract formalization of an informal description is developed. It will be used as the basis for the following phases. In this step, specifications can be formalized as either trace or functional specifications. The transition between these paradigms described by the Focus methodology is formally sound and correctness preserving. In Focus, the specifications are based on denotational semantics which map specifications onto sets of traces or sets of stream processing functions. Choosing stream processing functions, each function describes a possible behavior.

In the *design phase*, that follows the requirement phase, an essential part of the system development is carried out by developing the architecture of the system and refining it up to the required level of granularity. For the development of the specifications during the design phase, paradigms like functional and relational specifications as well as several specification styles like "Assumption/Commitment" or equational specifications are supplied using an uniform functional semantics. Due to the specific natures of these variants they can be used tailor-made for the solution of specific problems. During the last step, the *implementation phase*, the specifications, developed during the design phase, are transformed into one of the supplied implementation languages.

During the formal development of distributed systems the intended level of granularity is reached by stepwise refinement of the system. For this purpose Focus offers a powerful compositional refinement concept as well as refinement calculi, including behavioral, interface and structural refinement. On the whole, Focus offers a mathematically and conceptually sound framework supporting a complete system development from the abstract requirement specification to the implementation.

Since the semantic foundations of Focus including its development techniques are already explored in depth, the emphasis for further developments lies on a better applicability of the methodology, especially by system developers less experienced in formal methods. For that purpose, additional wide-spread description techniques, (semi-)automatic and schematic proof support and tailor-made methods for specific application fields have to be offered. Several techniques for describing and specifiying systems (like tables, state or system diagrams, MSC-like event traces, the "Assumption/Commitment" style) were successfully integrated in the methodology. With AutoFocus tool-support for a system-development is already available, giving future case studies a new quality by offering appropriate editors, consistency checks, code generation and even simulation.

Case studies are an important and stimulating work for testing FOCUS in different application areas. FOCUS will be further improved, using the experience gained from the case studies described in this paper and future studies to come. We invite the reader to use FOCUS for the specification and development of distributed systems, and contact us for additional support using the FOCUS methodology.

FOCUS is described in detail in

> Manfred Broy, Ketil Stølen.
> FOCUS *on System Development.*
> Manuscript, to appear, Springer, 1997.

and also in

> M. Broy, F. Dederichs, M. Fuchs, T.F. Gritzner, R. Weber.
> *The Design of Distributed Systems - An Introduction to* FOCUS.
> SFB-Report 342/2-2/92 A, Technische Universität München, 1993.

More information about AUTOFOCUS can be found in

> F. Huber, B. Schätz, A. Schmidt, K. Spies.
> AUTOFOCUS - *A Tool for Distributed Systems Specification.*
> in: Bengt Jonsson, Joachim Parrow (editor): "Proceedings FTRTFT'96 - Formal Techniques in Real-Time and Fault-Tolerant Systems", 467-470, LNCS 1135, Springer, 1996.

2 The Case Studies

We present a summary of the case studies done with FOCUS. For every case study the description is split into three parts:

(i) a brief problem description,

(ii) comments on the formal treatment with FOCUS, and

(iii) references to the respective paper(s).

The enumeration of the studies is continued from the predecessing paper:

> M. Broy, M. Fuchs, T.F. Gritzner, B. Schätz, K. Spies, K. Stølen.
> *Summary of Case Studies in* FOCUS –
> *A Design Method for Distributed Systems.*
> SFB-Report 342/13/94 A, Technische Universität München, 1994.

2.22 Communication Busses

(i) The FOCUS development methodology is applied to the development of a hardware bus as a communication medium for hardware components. A requirement specification of the bus interface with the definition of conflicts and resolution strategies is given. The initial requirement is refined into hardware-close specifications of bus driver devices and signal lines using formal refinement steps. Futhermore, two variant techniques for conflict resolution are informally introduced and formally specified.

(ii) The emphasis of the case study lies on the treatment of hardware oriented development, using schematic and "Assumption/Commitment" based specifications of bit-level circuits. The case study uses a special instantiation of the FOCUS development methodolgy especially suited for hardware oriented design, using appropriate assumptions about the communication behavior (infinite traces, fixed delays, signal-orientation). Specifications are given using an "Assumption/Commitment" scheme. Different forms of refinement steps are carried out (structural refinement, interface refinement). Each refinement step is formally verified using A/C-rules and predicate logic.

(iii) Roberto della Mura.
Spezifikation von Busstrukturen - Eine Fallstudie in FOCUS.
Master's Thesis, Technische Universität München, 1994.

2.23 A Radio Control Module

(i) In this Master's thesis a specification for a radio control module is developed. This module controls the operating modes of a radio. It reacts to a sequence of input commands by sending output commands to the various lower-level components of the radio. As this work was done in cooperation with an industrial partner, this work emphasizes the use of tables to offer intuitive specifications to users not familiar with formal specifications. One of the aims of the work was to test the capabilties of formal methods in an industrial environment. Another aspect is the motivation and application of data- and interface-refinement rules in the design process.

(ii) The case study covers the design level of FOCUS. A first, abstract specification of the radio control module undergoes structural, data and interface refinement yielding components close to an already existing implementation of the module. The design begins with a very abstract specification: a single component receives commands on one channel, and outputs control commands on another channel. The exact structure of the control commands must not yet defined. In a structural refinement step, this component is then split into one processing and one memory component. Using data refinement, the commands sent to the radio's lower-level components are then described in more detail; finally, interface refinement is applied to split the control module's single output channel into two channels corresponding to two internal busses of the radio. The thesis presents all needed refinement rules, and all refinement steps are formally proven.

(iii) Max Breitling.
Formale Entwicklung eines Funkgerätesteuermoduls.
Master's thesis, Technische Universität München, 1995.

2.24 A Modulo-N-Counter

(i) The use of functional system specifications and refinement in the formal development of hardware is illustrated using the example of a asynchronous Modulo N-counter device. The development includes modular specification, refinement and verification. The development is started at an intuitive requirements specification and is refined into a nontrivial concrete bit-level implementation.

(ii) The case study covers the complete range of the formal FOCUS development methodolgy. The development includes modular specification, refinement and verification. The refinement steps comprise behavioral, structural and interface refinement. The emphasis of this study is laid upon the modelling at different levels of abstraction and the verification conditions obtained by the refinement relations between these versions.

(iii) Maximilian Fuchs.
Formal Design of a Modulo-N Counter.
SFB-Report SFB 342/06/95 A, Technische Universität München, April 1995.

2.25 Abracadabra Service and Protocol

(i) The ISO/OSI basic reference model contains a variety of services and protocols in order to guarantee a secure data transfer. A well known service is the Abradadabra service which is realized by the Abracadabra protocol. The Abracadabra service guarantees a connection-oriented, symmetrical and reliable data exchange over an unreliable medium. In the Master's thesis, based on some ISO publications of informal and formal descriptions of the Abracadabra-example using techniques like LOTOS, SDL and Estelle, a formal specification of the Abracadabra service and protocol is developed within the FOCUS framework.

(ii) The informal description of the Abracadabra service and protocol is transformed in a formal specification using pulse-driven functions. The main emphasis lies on the demonstration of the suitability of the "Assumption/Commitment" style in modelling especially the connection and disconnection of components. In addition this Master's thesis shows that the development of a detailed formal specification discovers some lack of clarity in the informal description. Also some mistakes in the SDL specification, referenced by the ISO, are discovered and discussed. Modelling the Abracadabra protocoll was the first larger case study using the time dependent and pulse-driven FOCUS-semantics.

(iii) Christine Klein.
Spezifikation eines Dienstes und Protokolls in FOCUS
Master's Thesis, Technische Universität München, 1995

2.26 A Lift Controller

(i) Two different versions of a lift controller are covered in this case study. In the first version only one elevator cabin is considered, whereas for the second version the system is expanded to n lifts. In both examples there are m floors, each of them with two buttons: one for signaling the request to go up and one to go down (except on the top and the bottom floor). Each lift has a set of m buttons for the user to signal a request to transport him to a certain floor. In order to serve such requests — both internal (i.e. signaled from inside the cabin) and external (i.e. signaled from one of the floors) requests — a lift may perform a sequence of discrete actions like "up by one floor", "down by one floor" and "stop", with a "stop" in between each change of direction. The strategy used for these examples is the same as in example 2.7 (of part I) "A Lift Controller".

This case study covers the first two development levels while emphasing on the requirement specification.

(ii) The informal descriptions of both examples are translated into a formal trace specification, which is global, i.e. the system is not yet divided into seperate components like elevators or floors. Because of the little differences between the two specifications only the single lift system is transformed into a local trace specification. In order to do this it is necessary to incorporate a model of time. Two possible distribution are then considered: one with the lift and the m floors as the components, and the other one which has an additional component to control the communication between the lift and the floors. The later one is finally converted to a functional specification, which is the first step on the design level.

(iii) Jürgen Rudolph.
Entwicklung einer Liftsteuerung – Fallstudie in FOCUS.
Master's Thesis, Technische Universität München, 1995.

2.27 The Production Cell

(i) This study extends the work in 2.18, where a design specification of a production cell is developed, for the usage of mechanical proof support. The production cell works as described in 2.18: It consists of a press for metal plates and several components for handling the plates automatically. The work pieces are moved into the cell by a conveyor belt. An elevating table lifts them into an appropriate grasp position for a robot. The robot both moves plates waiting at the table into the press and also puts the pressed plates on a deposit belt.

This work concentrates on the development of some component specifications within a theorem prover environment. Starting with an abstract specification for the production cell, specifications for the subcomponents of the elevating table are developed by applying hierarchical refinement steps. Each refinement step is proved to be correct using the generic theorem prover *Isabelle*.

(ii) The emphasis of this case study is to support the proofs involved in the design phase by the logic HOLCF (Higher Order Logic of Computable Functions). HOLCF is itself based on the generic theorem prover *Isabelle*. The network description language ANDL is used to formalize FOCUS-agents in HOLCF. Also, FOCUS' refinement rules for "Assumption/Commitment" specifications are both formalized and proved to be correct in *Isabelle*/HOLCF. The proofs follow closely those in 2.18. However, while in 2.18 they are done as paper proofs, they are carried out completely within *Isabelle's* logical framework here. Especially the treatment of "Assumption/Commitment" specifications in *Isabelle* is of special interest.

(iii) (a) Robert Sandner.
Unterstützung von Strukturverfeinerungen in FOCUS *durch Isabelle – Verifikation einer Fertigungszelle.*
Master's Thesis, Technische Universität München, 1996.

(b) Robert Sandner, Olaf Müller.
Theorem Prover Support for the Refinement of Stream Processing Functions.
in: Ed Brinksma (editor): "Tools and Algorithms for the Construction and Analysis of Systems", LNCS 1217, Springer 1997.

2.28 A Buffer of Length One

(i) This case study illustrates the use of the functional system specifications and their refinement in the development of system components like it's explained in the FOCUS-methodology.

The paper presents the specification of a simple one element buffer as a component that can store one data element and return it upon request; of a fair loose one element buffer that may fail in storing data elements or in serving requests. It is fair in the sense that it will not fail forever on repeated attempts. The driver composed with the fair loose buffer behaves like a one element buffer.

The real time one element buffer is a one element buffer that operates in a discrete time frame. The real time fair loose one element buffer operates in a discrete time frame too. It indicates success by a positive acknowledgement within a fixed amount of time. The real time driver operates in a dircrete time frame. Composed with the real time fair loose buffer it behaves like the real time buffer.

(ii) The small example is used to demonstrate functional formalisms for the specification, refinement and verification of system components. The emphasis is put on the usefulness of functional formalisms in the development process and on the achieved modularity of the system descriptions and the development process.

The one element buffer, the loose buffer end the driver are each presented in three specifications using and demonstrating three specification styles: equational specifications, the "Assumption/Commitment" specification format and state-based specifications.

The system composed of the driver and the loose buffer is specified and the verification that this system is a refinement of the buffer is presented.

The specification of the real time components are presented and it is shown that these components are refinements of the components without time.

Finally the verification that the composition of the real time driver and loose buffer is a refinement of the real time buffer is given and it is shown that this verification can be concluded from the refinement steps concerning the timed components.

(iii) Manfred Broy.
Specification and Refinement of a buffer length one.
in: M. Broy (editor), "Deductive Program Design", 273-304, Springer, 1996, NATO ASI Series, Vol. 152.

2.29 Banking System

(i) The banking system consists of a network of banks interacting with each other and with the environment. Banks can be dynamically founded or liquidated. Each bank maintains a set of private accounts. Accounts can also be dynamically opened or deleted. A given amount of money can be deposited on an account, withdrawn from an account or transfered form one account to another account. Consistency checks should assure that money is withdrawn only if available on the corresponding account and transfered only to existing accounts. Time constrains should assure that given requests are processed in given time. This example shows the typical structure of a distributed information system as it usually occurs in practical applications: local conventional entities interact by exchanging messages.

(ii) Formally the following aspects had to be mastered:

(1) Mobility: interaction between the system components dynamically changes the system-configuration. Privacy preserving properties should control the amount of interference.

(2) Recursion: both banks and accounts can be dynamically created.

(3) Timing: requests are to be processed in a given amount of time.

(4) State: the interplay between classical data modeling aspects and interaction.

The formalism used to master these aspects is based on the model for mobile networks (also see the case study 2.33).

(iii) Available as slides only via WWW under

`http://www4.informatik.tu-muenchen.de/focus`

2.30 Video on Demand

(i) This article argues that formal techniques are indeed useful for practical application, but they should be put to indirect use. To demonstrate the approach, a formal semantics is defined for two forms of popular graphical description formalisms. Based on the formal definition, “safe” development steps and their graphical counterparts are introduced. This yields a graphical development method which relies on precise formal foundations.

(ii) To demonstrate the approach, two pragmatic graphical description techniques, taken from the field of telecommunication, are analyzed regarding their information content and their application in the process of specification development; as a result these techniques are formally defined using trace semantics and a clausal representation scheme. In a next step, refinement schemes for these graphical description techniques are introduced. Their refinement character is verified using their underlying formal semantics defined in the first step. The transformation schemes are expressed in a way suitable for a tool supported development. Together with a guide line for the application of the description techniques and the schemes, thus a tool supported development method for protocol oriented system requirement design is introduced.

Finally, possible further enhancements of the formally based development of technical systems using user oriented description techniques are discussed. The mentioned points include relaxations of the restrictive transformtion based development.

(iii) Manfred Broy, Heinrich Hußmann, Bernhard Schätz.
Graphical Development of Consistent System Specification.
in: M.-C. Gaudel, J. Woodcock (editors), "FME'96: Industrial Benefit and Advances in Formal Methods", 248-267, LNCS 1051, Springer, 1996.

2.31 Remote Procedure Call (using relations)

(i) A memory is specified and decomposed into subcomponents. Firstly, the overall behavior of the memory is specified. Both a reliable and an unreliable version is characterized. Secondly, the unreliable memory is decomposed into three component specifications, namely the specification of the reliable memory, and specifications of a RPC component and a server. The correctness of this decomposition is verified. Thirdly, the specification of the RPC component is decomposed into a lossy RPC component and a server. These two components communicate in accordance with a protocol based on timeouts. Also the correctness of this decomposition is verified.

(ii) A relational specification technique is used. Specifications are expressed in two different formats. The time-independent format is used to describe components whose behavior is time-independent. If time constraints are to be imposed, the time-dependent format is used. The emphasis is put on finding the right balance between readability and brevity. In particular, nondeterministic behavior is captured in terms of prophecies. The proofs are not formal.

(iii) Ketil Stølen.
Using Relations on Streams to Solve the RPC-Memory Specification Problem.
in: M. Broy, S. Merz, K. Spies (editors), "Formal Systems Specification – The RPC-Memory Specification Case Study", 477-520, LNCS 1169, Springer 1996.

2.32 Remote Procedure Call (functional solution)

(i) see the RPC-description of case study 2.31.

(ii) A functional specification of the syntactic interface and the black box behavoir of the unreliable and the reliable memory component and the RPC-compontent is given. Additionally the clerk for driving the RPC component is specified. These components are separately specified in a modular way. A short discussion of these specifications is presented and their composition into a distributed system of interacting components in a modular way is shown. The proof that the specification of the composed system fulfills again the requirement specification of the reliable memory component is given. Finally a timed version of the RPC component and of a clerk component and their composition is presented.

(iii) Manfred Broy.
A Functional Solution to the RPC-Memory Specification Problem.
in: M. Broy, S. Merz, K. Spies (editors), "Formal Systems Specification – The RPC-Memory Specification Case Study", 183-211, LNCS 1169, Springer 1996.

2.33 Modelling Mobile and Dynamic Systems – A Simple Batchsystem

(i) The basic semantics of Focus does not support the modelling of mobile and dynamic systems. Mobile systems covers interaction between components which may change their communication partners during a system run. Dynamic systems allow the creation of new components and the deletion of existing components during the life of the network. A simple example for dynamic, mobile systems is represented by the batch system.

In its initial configuration, the batch system consists of one master component receiving orders from a user in the environment. The orders are distributed to some submaster components which are created by the master component. For each order (task) a submaster itself creates slave components each handling one task. Due to a given capacity of the submaster only a limited number of slaves can be created. If the maximum number of slaves is created, the master component is informed by the submaster. Receiving a new order by the environment the master component creates the next new submaster.

(ii) This case study focuses on the development of a method for the specification of dynamic, mobile systems. First the semantic model for mobile, dynamic systems is introduced with regard to users who are versed in the semantic model of Focus and interested in learning its extension for mobile and dynamic systems. Then, specifying the batch system, it is demonstrated how the specification is developed in a systematic and step-wise way. Using pulse-driven function and a constructive and operational specification style with recursive function equations, guidelines and pattern-like formulas are introduced allowing users to specify mobile, dynamic behaviour in a uniform and schematic way.

(iii) (a) Ursula Hinkel, Katharina Spies.
Anleitung zur Spezifikation von mobilen, dynamischen FOCUS-*Netzen.*
SFB-Report 342/16/96 A, Technische Universität München, 1996.

(b) Ursula Hinkel, Katharina Spies.
Spezifikationsmethodik für mobile, dynamische FOCUS-*Netze*
GI/ITG-Fachgespräch 1997, Berlin.
Adam Wolisz, Ina Schieferdecker, Axel Rennoch (Hrsg.)
GMD-Studie Nr. 315, GMD-Forschungszentrum, 1997.

2.34 Secure Systems

(i) As part of the development of a methodology covering the formal analysis of system security, an authentic and available server component is being developed. Starting from a system specification which can be shown not to be authentic, security mechanisms based on simple cryptographic authentication protocols are introduced. Security analysis works out the subtle differences between two protocol variants based on ISO 9798-2 (challenge-response with encrypted response) and ISO 10181-2 (challenge-response with encrypted challenge), and with respect to different adversary models. The authenticity proof of the ISO 9798-2 variant leads to an improvement considering mechanism embedment which also provides availability if certain fairness conditions on the adversary behavior hold.

(ii) The purpose of the case study is to show the usefulness of a methodology for the formal development of secure systems in FOCUS which is based on explicit adversary modelling and defining security as a relation between two system specifications describing ordinary system behaviour and attack situations. The methodology is based on pulse-driven stream-processing function semantics. The case study shows the use of different FOCUS specification styles and formats on the design level. When introducing the security protocols, a functional specification can be immediately achieved from the informal description given in the ISO standards documents. The functional specification turns out to be cumbersome with respect to security proofs; thus, a relational variant is given. The functional specification is shown to refine the relational one, which itself refines the original system specification, showing that the introduction of the security mechanisms does not violate the intended system behavior. Based on the relational specification, authenticity properties are proved with respect to a simple and a more complex adversary model. With the authenticity proof, it turns out that due to the buffering of incoming challenges, the system is blocked in case of an attack, with availability being lost. In order to achieve availability, the timing of challenges has to be considered, leading to a time dependent protocol specification that is available with respect to some fairness assumptions and preserves authenticity.

(iii) (a) Volkmar Lotz.
Threat Scenarios as a Means to Formally Develop Secure Systems.
in: E. Bertino, H. Kurth, G. Martella, E. Montolivo (editors), "Computer Security – ESORICS '96", LNCS 1146, Springer, 1996.

(b) Volkmar Lotz.
Threat Scenarios as a Means to Formally Develop Secure Systems.
Technical Report TUM-I9709, Technische Universität München, 1997.

2.35 A Traffic Light Controller

(i) Starting from an informal textual description, a traffic light controller of a simple two-way intersection is specified by a network of data flow components. The main focus of this work is in the use of readable description techniques, such as state transition diagrams and tables, to hide the mathematical background of the specifications. A second main point is the use of automated formal verification tools to prove properties of the specification.

(ii) The specification consists of three components that communicate via synchronous streams, where in each time interval exactly one message is transmitted. Two components handle the low-level interfaces to the inductive loop sensors and the signal lamps. They are specified as a relation between input and output streams, and as a small functional program, respectively. The third component is the controller itself. It is modeled as a state transition diagram, where the individual transitions are described with tables. This makes the specification comparatively compact. The specification is validated by translating it to a program in the input language of the model checker SMV, and verifying formulas in the temporal logic CTL that describe certain aspect of the expected controller behavior.

(iii) Jan Philipps, Alexander Schmidt.
Traffic Flow by Data Flow.
SFB-Report 342/13/97A, Technische Universität München, 1997.

2.36 The TimeWarp Simulator

(i) The Time Warp-mechanism accomplishes an efficient synchronisation between the components of a distributed discrete event-driven simulator. According to this mechanism the components of the simulator calculate locally while sending results to other components without waiting for any events to be received, with the optimistic hope that causality is preserved. In case of a miscalculation recognized by messages received too late, the components have to perform a rollback and cancel some messages already sent, maybe causing other components also to rollback. Nevertheless the distributed Time Warp-algorithm returns a correct result. In this case study this technique is modelled with the development method Focus and the correctness is formally investigated. Starting from a simple, centralized simulator three development steps are performed, reaching a distributed simulator using Time Warp. The simulators on various abstraction levels are specified formally and the development steps are verified using the techniques of Focus.

(ii) In this case study, a distributed simulator was developed through four different abstraction layers. The validity of the refinements (including behaviour, structural and interface refinement) is proved formally, showing that the communication mechanism of the Time Warp is correct if the algorithm for the calculation of the global virtual time is asumed to be correct.

An extensive multiple use of similar components occures, making the need for appropriate notions obvious. A suggestion for tabular specifications was given that uses a variable number of communication channels, described by parametrized columns. Also a tabular form for describing the communication behaviour for synchronous communication was suggested. Some ideas that could lead to an improved support for the treatment of refinement relations can be found in the detailed proofs, e.g. proofs using the states of systems, using equivalence classes of the streams and using the concept of refinement mappings.

(iii) Max Breitling.
Specifying and Verifying TimeWarp *with* Focus.
To appear, Technische Universität München, 1997.

2.37 The Steam Boiler

(i) This paper demonstrates how to carry out the specification of a steam boiler and its controller in Focus. We do not give a complete detailed specification, but concentrate on the conceptual formal model in Focus for the control task starting from classical control theory. We apply Focus to the development of the requirement specification of a steam boiler controller to demonstrate some of its strengths: its expressiveness and adaptability in different domains – here the adaptation to control theory –, its user-friendly extensions – here the table notation technique – and the broad range of refinement techniques – here the concept of data and state refinement.

(ii) In the first part of the article the terms and the functional model of control theory are mapped on the semantic model of stream processsing functions. A network scheme is introduced capturing the basic notions of control theory. In the second part the notion of structured specifications using conceptual states and a special form of tabular representation for the description of these system requirements is introduced. Again, stream processing functions are used as the formal model; thus, tables are interpreted as short-hand notations for predicates over stream processing functions and with states as additional parameter. To demonstrate the possibilities and restrictions of such an approach, several requirements of the steam boiler are represented in tabular fashion.

Finally, refinement is discussed and embedded in the tabular approach and demonstrated in the steam boiler context.

(iii) M. Broy, F. Regensburger, B. Schätz, K. Spies.
Streams of Steam.
SFB-Report 342/14/97A, Technische Universität München, 1997.

Acknowledgements

We would like to thank Radu Grosu, Ursula Hinkel, Volkmar Lotz, Jan Philipps, Jürgen Rudolph and Robert Sandner for their contributions to this paper.

SFB 342: Methoden und Werkzeuge für die Nutzung paralleler Rechnerarchitekturen

bisher erschienen :

Reihe A

Liste aller erschienenen Berichte von 1990-1994
auf besondere Anforderung

342/01/95 A	Hans-Joachim Bungartz: Higher Order Finite Elements on Sparse Grids
342/02/95 A	Tao Zhang, Seonglim Kang, Lester R. Lipsky: The Performance of Parallel Computers: Order Statistics and Amdahl's Law
342/03/95 A	Lester R. Lipsky, Appie van de Liefvoort: Transformation of the Kronecker Product of Identical Servers to a Reduced Product Space
342/04/95 A	Pierre Fiorini, Lester R. Lipsky, Wen-Jung Hsin, Appie van de Liefvoort: Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes
342/05/95 A	Sascha Hilgenfeldt, Robert Balder, Christoph Zenger: Sparse Grids: Applications to Multi-dimensional Schrödinger Problems
342/06/95 A	Maximilian Fuchs: Formal Design of a Model-N Counter
342/07/95 A	Hans-Joachim Bungartz, Stefan Schulte: Coupled Problems in Microsystem Technology
342/08/95 A	Alexander Pfaffinger: Parallel Communication on Workstation Networks with Complex Topologies
342/09/95 A	Ketil Stølen: Assumption/Commitment Rules for Data-flow Networks - with an Emphasis on Completeness
342/10/95 A	Ketil Stølen, Max Fuchs: A Formal Method for Hardware/Software Co-Design
342/11/95 A	Thomas Schnekenburger: The ALDY Load Distribution System
342/12/95 A	Javier Esparza, Stefan Römer, Walter Vogler: An Improvement of McMillan's Unfolding Algorithm
342/13/95 A	Stephan Melzer, Javier Esparza: Checking System Properties via Integer Programming
342/14/95 A	Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Point-to-Point Dataflow Networks
342/15/95 A	Andrei Kovalyov, Javier Esparza: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs

Reihe A

342/16/95 A Bernhard Schätz, Katharina Spies: Formale Syntax zur logischen Kernsprache der Focus-Entwicklungsmethodik

342/17/95 A Georg Stellner: Using CoCheck on a Network of Workstations

342/18/95 A Arndt Bode, Thomas Ludwig, Vaidy Sunderam, Roland Wismüller: Workshop on PVM, MPI, Tools and Applications

342/19/95 A Thomas Schnekenburger: Integration of Load Distribution into ParMod-C

342/20/95 A Ketil Stølen: Refinement Principles Supporting the Transition from Asynchronous to Synchronous Communication

342/21/95 A Andreas Listl, Giannis Bozas: Performance Gains Using Subpages for Cache Coherency Control

342/22/95 A Volker Heun, Ernst W. Mayr: Embedding Graphs with Bounded Treewidth into Optimal Hypercubes

342/23/95 A Petr Jančar, Javier Esparza: Deciding Finiteness of Petri Nets up to Bisimulation

342/24/95 A M. Jung, U. Rüde: Implicit Extrapolation Methods for Variable Coefficient Problems

342/01/96 A Michael Griebel, Tilman Neunhoeffer, Hans Regler: Algebraic Multigrid Methods for the Solution of the Navier-Stokes Equations in Complicated Geometries

342/02/96 A Thomas Grauschopf, Michael Griebel, Hans Regler: Additive Multilevel-Preconditioners based on Bilinear Interpolation, Matrix Dependent Geometric Coarsening and Algebraic-Multigrid Coarsening for Second Order Elliptic PDEs

342/03/96 A Volker Heun, Ernst W. Mayr: Optimal Dynamic Edge-Disjoint Embeddings of Complete Binary Trees into Hypercubes

342/04/96 A Thomas Huckle: Efficient Computation of Sparse Approximate Inverses

342/05/96 A Thomas Ludwig, Roland Wismüller, Vaidy Sunderam, Arndt Bode: OMIS — On-line Monitoring Interface Specification

342/06/96 A Ekkart Kindler: A Compositional Partial Order Semantics for Petri Net Components

342/07/96 A Richard Mayr: Some Results on Basic Parallel Processes

342/08/96 A Ralph Radermacher, Frank Weimer: INSEL Syntax-Bericht

342/09/96 A P.P. Spies, C. Eckert, M. Lange, D. Marek, R. Radermacher, F. Weimer, H.-M. Windisch: Sprachkonzepte zur Konstruktion verteilter Systeme

342/10/96 A Stefan Lamberts, Thomas Ludwig, Christian Röder, Arndt Bode: PFSLib – A File System for Parallel Programming Environments

342/11/96 A Manfred Broy, Gheorghe Ştefănescu: The Algebra of Stream Processing Functions

342/12/96 A Javier Esparza: Reachability in Live and Safe Free-Choice Petri Nets is NP-complete

342/13/96 A Radu Grosu, Ketil Stølen: A Denotational Model for Mobile Many-to-Many Data-flow Networks

342/14/96 A Giannis Bozas, Michael Jaedicke, Andreas Listl, Bernhard Mitschang, Angelika Reiser, Stephan Zimmermann: On Transforming a Sequential SQL-DBMS into a Parallel One: First Results and Experiences of the MIDAS Project

Reihe A

342/15/96 A Richard Mayr: A Tableau System for Model Checking Petri Nets with a Fragment of the Linear Time μ -Calculus
342/16/96 A Ursula Hinkel, Katharina Spies: Anleitung zur Spezifikation von mobilen, dynamischen Focus-Netzen
342/17/96 A Richard Mayr: Model Checking PA-Processes
342/18/96 A Michaela Huhn, Peter Niebert, Frank Wallner: Put your Model Checker on Diet: Verification on Local States
342/01/97 A Tobias Müller, Stefan Lamberts, Ursula Maier, Georg Stellner: Evaluierung der Leistungsf"ahigkeit eines ATM-Netzes mit parallelen Programmierbibliotheken
342/02/97 A Hans-Joachim Bungartz and Thomas Dornseifer: Sparse Grids: Recent Developments for Elliptic Partial Differential Equations
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 gic
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
342/12/97 A Christian B. Czech: Architektur und Konzept des Dycos-Kerns
342/13/97 A Jan Philipps, Alexander Schmidt: Traffic Flow by Data Flow
342/14/97 A Norbert Fröhlich, Rolf Schlagenhaft, Josef Fleischmann: Partitioning VLSI-Circuits for Parallel Simulation on Transistor Level
342/15/97 A Frank Weimer: DaViT: Ein System zur interaktiven Ausführung und zur Visualisierung von INSEL-Programmen
342/16/97 A Niels Reimer, Jürgen Rudolph, Katharina Spies: Von FOCUS nach INSEL - Eine Aufzugssteuerung
342/17/97 A Radu Grosu, Ketil Stølen, Manfred Broy: A Denotational Model for Mobile Point-to-Point Data-flow Networks with Channel Sharing
342/18/97 A Christian Röder, Georg Stellner: Design of Load Management for Parallel Applications in Networks of Heterogenous Workstations
342/19/97 A Frank Wallner: Model Checking LTL Using Net Unfoldings
342/20/97 A Andreas Wolf, Andreas Kmoch: Einsatz eines automatischen Theorembeweisers in einer taktikgesteuerten Beweisumgebung zur Lösung eines Beispiels aus der Hardware-Verifikation – Fallstudie –

Reihe A

342/21/97 A	Andreas Wolf, Marc Fuchs: Cooperative Parallel Automated Theorem Proving
342/22/97 A	T. Ludwig, R. Wismüller, V. Sunderam, A. Bode: OMIS - On-line Monitoring Interface Specification (Version 2.0)
342/23/97 A	Stephan Merkel: Verification of Fault Tolerant Algorithms Using PEP
342/24/97 A	Manfred Broy, Max Breitling, Bernhard Schätz, Katharina Spies: Summary of Case Studies in Focus - Part II

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 Paech1: 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