Report on the Dagstuhl - Seminar 9619

Semantics of Concurrent Systems - Foundations and Applications

6 - 10 May 1996

Organisers:

Manfred Droste,	Technical University of Dresden, Germany
Ernst-Rüdiger Olderog,	University of Oldenburg, Germany
Bernhard Steffen,	University of Passau, Germany
Glynn Winskel,	Aarhus University, Denmark

Introduction

Many real computer systems exhibit concurrrency, i.e. they consist of a number of components that work largely independently but also interact with each other. Such concurrent systems are difficult to design because concepts like causality, nondeterminism, synchronization, communication and timing constraints require an elaborate refined treatment which, in its complexity, can only be handled on a formal basis. This led to an intensive research into formal mehods for the specification, design, verification and analysis of concurrent systems. At the heart of these mehods are semantic models, tailored for the treatment of particular aspects of concurrent systems. We observe two directions in this research.

The investigation of *semantic foundations* of concurrent systems, which is characterized by emphasizing mathematically tractable semantic domains equipped with suitable operators for system composition. Often this poses mathematically interesting problems in domain theory. To solve them, concurrent systems are studied at a high level of abstraction. For example, uninterpreted action symbols rather than state transforming actions are considered.

The *application* of given semantic models to obtain logics, algorithms and calculi for the specification, verification and design of concurrent systems. Typically this work is driven by concrete case studies, which often reveal the need for changes or at least adaptations of the known abstract semantic models for concurrency.

The aim of the proposed Dagstuhl seminar was the discussion of semantic foundations of concurrent systems in the light of their applicability for the verification and design of concurrent systems. This concerned in particular the impact of the mathematical structures of the semantic domains on the compositionality of the various verification methods.

Particular topics of interest were:

- new developments in the semantics of concurrency,
- the relationship between different semantic models,
- semantic based approaches to the verification and design of concurrent systems,
- the suitability of specific semantic models for the task of reasoning about concurrent systems.

The programme of the seminar was intense and stimulating; it comprised 32 talks and one evening discussion on the impact of semantics. The abstracts of the talks are recorded in this report in the order of presentation.

The evening discussion reviewed visible achievements of semantics like the use of model checking and abstract interpretation, the design of functional and synchronous languages, the influence of Hoare's CSP on Ada and OCCAM, the clarification of various concepts concerning the computational process like interleaving versus partial order semantics, the use of types, fixed points and invariants. The renewed interest in logics and category theory was cited amongst the scientific impacts of semantics.

As new research areas for semantics the following were recommended: security, performance analysis, the integration of semantics with complexity, hardware design languages and hybrid systems.

Abstracts

Willem-Paul de Roever, Christian-Albrechts-University Kiel, Germany
Compositionality in Real-Time Shared Variable Concurrency

Whereas for distributed communication a multitude of, even compositional, proof systems for real-time exist, for real-time shared variable concurrency the picture is quite bleak. This is the more astonishing because it can be argued that hard real-time systems are based on shared variable concurrency. The papers by Abadi & Lamport '91 on incorporation of this feature within TLA, and by Schneider, Bloom & Marzullo '92 using a noncompositional method (proof outlines) form notable exceptions. In the present paper we present a compositional Hoare style proof system for timed exclusive write and multiple read accesses to shared memory which is based on a compositional semantics which is fully abstract w.r.t. an obvious operational semantics, hence containing the minimal amount of information for its characterization. Moreover this axiomatization is sound and complete. It is illustrated on a correctness proof of a real-time mutual exclusion attributed to M. Fisher and carried out within the framework of a formalization of our theory within PVS, hence allowing machine support for our proofs.

Joachim Parrow, KTH, Sweden
Trios in Concert

A trio is a term $\alpha.\ \beta.\ \gamma.\ 0$ in the polyadic π - calculus. We show that restricted parallel composition of possibly replicated trios is enough to obtain the expressive power of the full summation-free π - calculus. Therefore that fragment of the calculus is undecidable.

Matthew Hennessy, University of Sussex, Great Britain
A Theory of Weak Bisimulation for Core CML
(Joint work with W. Fereira and A. Jeffrey)

Concurrent ML is an extension of standard ML of New Jersey with concurrent features similar to those of process algebra. Reppy has given it an operational semantics based on reductions of configurations, using entire programs rather than program fragments.

Here we give a compositional operational semantics for a fragment of CML, in terms of a labelled transition system where the nodes are program fragments. We then use this labelled transition system to define a notion of weak bisimulation equivalence, based on higher-order bisimulations. We prove that this is preserved by all program contexts and give some examples which indicates that the resulting theory generalises that for process algebras such a CCS and the standard sequential theory of ML.

Ugo Montanari, University of Pisa, Italy
History-Dependent Automata
(Joint work with Marco Pistore)

In the talk we introduce history-dependent automata, i.e. automata whose states and transitions are enriched with local names, and define a bisimulation equivalence on them. Our aim is to show that history-dependent automata represent a good operational model for those calculi whose labels refer to names declared by previous transitions; as a case study we consider the π-calculus. We also give an equivalent categorical definition of history-dependent automata (as internal pointed, labelled graphs on a category of named sets) and of bisimulation equivalence on them (via open maps).

Burghard v. Karger, Christian-Albrechts-University Kiel, Germany
Temporal Algebra

Temporal logic and the theory of Galois connections are two gems of theoretical computer science which are rarely, if ever, mentioned together. One purpose of this talk is to show that they may be seen as two sides of the same coin.

We develop temporal logic from the theory of complete lattices, Galois connections, and fixed points. In particular, we prove that all seventeen axioms of Manna and Pnueli's sound and complete proof system for linear temporal logic can be derived from just two postulates, namely,

(O, O) is a Galois connection

(O, O) is a perfect Galois connection

In the second part of the talk we investigate relational models of temporal logic. The main result is a discretization theorem which allows us to reuse the theory of discrete temporal algebras for the continous case.

The papers "Temporal Logic via Galois Connections" and "A relational model for temporal logic" are available from the author's homepage http://www.informatik.uni-kiel.de/~bvk/

Jaco W. de Bakker, CWI, Amsterdam, The Netherlands
Semantics for Unguarded Recursion

So far, semantic models based on metric spaces (cf. [BV]) have postulated that recursive procedures be guarded (each occurrence of a procedure variable in a procedure body is sequentially preceded by an elementary action). Mathematically, this leads to contractivity of the operator associated with a procedure body, allowing us the use of unique fixed points by Banach's theorem. In this talk, we explore the consequences of dropping the guardedness requirement, using a simple language combining recursion and concurrency as a vehicle. Denotationally, it turns out that it is convenient to use greatest rather than unique fixed points (as already noted in [N]). Operationally, more work needs to be done, including an analysis of the various types of unguardedness, the addition of several new rules to the transition system specification, and a more complex higher-order definition of the semantics induced by the new system. In the equivalence proof relating the operational and denotational semantics, we have to stretch the limits of the Kok/Rutten proof method, also exploiting some results from classical semantics for recursion.

[BV] J.W. de Bakker and E.P. de Vink, Control Flow Semantics,
MIT Press, 1996.

[N] M. Nivat, Infinite words, infinite trees, infinite computations, in
J.W. de Bakker, J. van Leeuwen, eds., Foundations of Computer Science,
Vol. III.2, Mathematical Centre Tracts 109, pp. 3-52, 1979.

Reinhold Heckmann, University of Saarland, Germany
Partial Metric Spaces

In the paper, "A Computational Model for Metric Spaces", Abbas Edalot (Imperial College) and I defined a continuous poset BX of "formal balls" for every metric space X. The subspace of maximal elements of BX is homeomorphic to X. The poset BX is directed complete iff the metric space X is complete. This suggests the following completion procedure for a metric space X: from BX, do rounded ideal completion to obtain I(BX), and restrict to the maximal elements. In this moment, the metric completion is obtained as a topological space. To obtain its metric as well, the metric on X has to be extended to a Scott continous distance function p on BX which can be made subject to rounded ideal completion.

A suitable choice of such a function p is given. It satisfies all but one axiom of a partial metric space in the sense of Matthews so that we call it partial metric again. Partial metrics differ from proper metrics in that self-distances may be different from 0. Every partial metric space induces a topology, which in case of BX is the Scott topology. The partial metric spaces BX are canonical in the sense that arbitary partial metric spaces can be embedded into a BX-space.

We conclude with some examples. There is a nice partial metric on streams which creates the Scott topology w.r.t. prefix ordering. In contrast, we did not succeed to find a partial metric on the continuous dcpo of non-empty compact subsets of IR^2 (ordered by "$\supseteq$") which creates the Scott topology and produces the usual distance when applied to two singletons. It is open whether such a partial metric exists.

Philippe Darondeau, University of Rennes, France
Stratified Sums of Nets
(Joint work with E. Badovel, INRIA-Rennes)

One may construct a general duality between pure nets and automata, parameterized on the type of nets. A type of nets is given by a deterministic automaton whose states range over the possible values for a place and whose transitions range over the possible evolutions of a place when a transition is fired in a net. Stratified sum of nets appear through this general duality as a counterpart to cascade products of automata. Stratified P/T-nets are a subclass of Valk's self modifying nets, with acyclic dependencies between places, and the decision of the synthesis problem for this class of nets takes polynomical time. Stratified 2-nets are sums of nets of type $\mathbb{Z}$, when $\mathbb{Z}$ is the Cayley graph of the boolean group. A finite complete automaton is isomorphic to the marking graph of a stratified 2-net iff its transformation group is a 2-group.

Stephen Brookes, Carnegie Mellon University, USA
The Essence of Parallel Algol

We consider a parallel Algol-like language, combining the λ–calculus with shared-variable parallelism. We provide a denotational semantics for this language, simultaneously adapting the possible worlds model of Reynolds & Oles to the parallel setting and generalizing the "transition traces" model for shared-variable programs to the procedural setting. This semantics supports reasoning about safety and liveness properties, and validates a number of natural laws of program equivalence. We also provide a relationally parametric semantics, generalizing the model to permit reasoning about relation-preserving properties of programs, and adapting work of O'Hara & Tennert to the parallel setting. This semantics supports standard methods of reasoning about representational independence. The clean design of the programming language and its semantics supports the orthogonality of procedures and shared-variable parallelism.

Eugene Stark, SUNY at Stony Brook, USA
Bifibrational Semantics of Process Network

The object of this talk is to motivate why bicategories of bifibrations are interesting and relevant for concurrency theory.

We present a model of dataflow-like process networks as systems of inequalities, together with a graphical syntax. We use this model to motivate the following conception of the study of network algebra: Given a locally posetal bicategory IB with cartesian structure, whose objects represent types whose arrows represent basic deterministic processes, and whose 2-cells represent information ordering, the "free network algebra" generated by IB should be a bicategory with the same objects as IB, whose arrows are certain bifibrations in IB, viewed as spans in IB, an whose 2-cells are certain arrows of spans, where bifibrations are composed via fibrational composite.

Javier Esparza, Technical University of Munich, Germany
An Improvement of Mc Millan's Unfolding Algorithm
(Joint work with Stefan Römer and Walter Vogler)

In a seminal paper at CAV '92, Mc Millan has proposed a new technique to avoid the state explosion problem in the verification of systems modelled by finite-state Petri nets. The technique is based on the notion of net unfolding. The unfolding of a net is another net, usually infinite but with a simpler structure. Mc Millan proposes an algorithm for the construction of a finite initial part of the unfolding which contains full information about the rechable state.

Mc Millan's algorithm is simple and elegant, but it may generate initial parts much longer than necessary, in the worst case exponentionally larger than the state space. In the talk we present a modification of the algorithm which avoids this problem. The initial part generated by the new algorithm is never larger than the state space.

Eike Best, University of Hildesheim, Germany
Coloured Nets with Curry

The objective of this work is to find a linear-algebraic calculus for Zemsens's coloured nets which
a) is a smooth generalisation from P/T-nets,
b) incorporates folding / unfolding in the same framework.

For details see the corresponding Technical Report (1996) or write to e.best@informatik.uni-hildesheim.de or visit www.informatik.uni-hildesheim.de

Michael Mislove, Tulane University, USA
A Category for Unbounded Nondeterminism and Thoughts on Full Abstraction

In this talk we consider whether there exist analogues to the traditional power domains which can support unbounded nondeterminism. From previous work with Roscoe & Schneider (TCS, 1995), we use the concept of a "local cpo" to generate a category of objects which consist of pairs (P, D) where P is a local cpo and D is a subcpo, and whose morphismus are pairs of montone maps (f,g), (P, D) → (Q, E) with f⊆y and g(D)⊆E. The "Dominated Convergence Theorem" assures f has a least fixed point. We show the category of these objects is closed under a number of the usual domain-theoretic constructors, and that, for any domain D, it has an analogue of the Smyth power domain which supports unbounded nondeterminism. The projection onto the category whose objects are local cpo's and whose morphismus are dominated monoton maps yields a cortesian closed category which has the relevant power domain over any domain, and the fixed point operator is again a morphism of the category.

At the end of the talk, we expressed the hope that results of the author and Frank Oles (TCS, 1995) showing full abstraction would extend to this setting.

Allan Cheng, Aarhus University, Denmark
Open Maps
(Joint work with Mogens Nielsen)

Spans of open maps have been proposed by Joyal, Nielsen, and Winskel as a way of adjoining an abstract equivalence, P-bisimilarity, to a category of models of computation M, where P is an arbitrary subcategory of observations. In this talk we briefly review the theory of open maps, we present results supporting the claim that P-bisimilarity is a general notion of bisimulation, we mention a few interesting observations we've encountered, and, finally, we propose a simple notion of a functor F:M →M being P-factorisable. Such functors always preserve P-bisimularity. Moreover, our proposed notion allows us to parametrise proofs of functors being P-factorisable with respect to the category of observations P, i.e., with respect ot a behavioural equivalence.

Glynn Winskel, Aarhus University, Denmark
Presheaf Models

We give a status report on presheaf models for process calculi and their use in modelling CCS-like languages as well as higher-order process calculi (late value-passing, process-passing and π-calculus).

Gerald Lüttgen, University of Passau, Germany
An Algebraic Approach to Distributed Priorities
(Joint work with Rance Cleaveland and V. Natarajan, N.C. State University)

In this talk we present a process algebra for distributed systems in which some actions may take precedence over others. In contrast with existing approaches to priorities, our algebra only allows actions to take priority over others at the same "location" and therefore captures a notion of localized preemption. Using Park's and Milner's notion of strong bisimulation as a basis, we develop a behavioral congruence and axiomatize it for finite processes. Also an associated observational congruence can be derived, and logical characterizations of the behavioral relations can be given. A simple example is presented that highlights the utility of the theory.

Paul Gastin, LITP-IBP, Universite Paris, France
Asynchronous Cellular Automata and Monadic Second Order Logic for Pomsets without Auto-Concurrency
(Joint work with Manfred Droste)

We extend to pomsets without auto-concurrency the fundamental notions of Asynchronous Cellular Automata (ACA) and of Asynchronous Mappings (AM) which were introduced for traces by Zielonka. Naturally the question of comparing the expressive power of AM, deterministic ACA, nondeterministic ACA and Monadic Second Order Logic (MSOL) for pomsets arises. For pomsets without auto-concurrency, we show that MSOL is at least as expressive as nondet-ACA and that det-ACA is at least as expressive as AM. The question whether the expressiveness is the same for AM, det-ACA, nondet-ACA and MSOL is open in general. However for a class of pomsets which satisfy a natural axiom we show that these "devices" have the same expressive power. The axiom essentially ensures that the ACA works as a Concurrent Read Exclusive Owner Write machine on the pomsets.

Igor Walukiewicz, Aarhus University, Denmark
Monadic Second Order Logic on Simple Models of Concurrency

In first part of the talk we review characterisations of recognisable sets of ω-words, ω-trees and ω -traces using Monadic Second Order Logic (MSOL) and fixpoint logics. We continue by considering MSOL over transition systems and show that if an MSOL formula defines a bisimulation closed set of transition systems then it is equivalent to a μ-calculus formula.

Next we ask the question, what classes of graphs have decidable MSOL theory. We present two operations on classes of graphs which preserve decidability of MSOL-theories. One operation is

Muchnik-Sheloh construction of creating trees of structures. The other one is Rabin's interpretation method adapted to MSOL by Coarcell and called MSOL-transduction.

In the final part of the talk we consider a problem of defining modal logic on (real) dependence graphs. To this end we define a function T which constructs a tree representation of a dependence graph such that both T and T^{-1} are MSOL transductions.

Manfred Droste, Technical University of Dresden, Germany
Concurrent Automata
(Joint work mostly with Dietrich Kuske, Dresden)

Automata with concurrency relations **A** are labelled transition systems equipped with a collection of binary relations describing when two actions in a given state of the automation can occur independently of each other. They arise naturally from place/transition systems with capacities from Petri net theory. The concurrency relations include a natural equivalence relation on the set of finite computation sequences of **A**. The associated equivalence classes, with composition as operation, form a monoid, the concurrency monoid M(**A**). Under suitable assumption on **A,** the elements of M(**A**) can be represented as labelled pomsets, so we can use logical formulae to specify languages in M(**A**). We show that a language in M(**A**) is recognizable iff it is definable by a sentence of monadic second order logic iff it is concurrently regular. Under specific assumptions on **A**, we obtain relationships between the classes of aperiodic, starfree and first order definable languages, resp., in M(**A**). This generalizes various recent results in trace theory.

Dietrich Kuske, Technical University of Dresden, Germany
A Temporal Logic for Computations of Concurrent Automata
(Joint work with Manfred Droste)

We generalize the result of Kamp that a temporal logic for words is expressively equivalent to the first order logic to the realm of concurrent automata. An automaton with concurrency relations **A** is a labelled transition system with a collection of binary relations describing when two actions in a given state of the automaton can occur independently of each other. These concurrency relations induce a natural equivalence on the set of finite and infinite computation sequences of **A**. The associated equivalence classes can be seen as parallel comutations of the automation **A**. For these computations we define a temporal logic, called TLPO with future and past tense operators. Furthermore, the computation of **A** can be represented by labelled partially ordered sets. We show that the temporal logic TLPO is expressively equivalent to the first order logic for labelled partially ordered sets. This generalizes recent results by Ebinger in trace theory.

Rob van Glabbeek, Stanford University, USA

Petri Nets. Configuration Structures, Propositional Theories and History Preserving Process Graphs.

Translations between several models of concurrency are reviewed cq. proposed. The models considered capture causality and branching time (and their interplay) and this behaviour is preserved by the translations.

Starting point is the work of Nielsen, Plotkin and Winskel, in which safe Petri nets are translated, through the intermediate steps of occurrence nets, prime event structures with a binary conflict relation, and their families of configuration, into a class of Scott domains. The resulting Scott domains, which were originally not intended to denote (separate) concurrent systems, can easily be regarded as transition systems, or process graphs. These graphs capture causality through confluence of squares. Therefore I like to see the connection between safe Petri nets and these domains as one between a class of nets and a class of process graphs. This gives rise to the question which process graphs capture causality through confluence of squares. My answer is 'all of them', and I propose an unfolding of arbitrary graphs into the so-called history preserving ones, preserving this structure. The history preserving process graphs generalize the Scott domains originating from prime, or even general event structures.

Several authors have proposed transition systems, enriched with some auxiliary structure to capture causality, as a model of concurrency, of the concurrent transition systems of Stark and Droste, the asynchronous transition systems of Shields and Bednardczyc and the transition systems with independence of Nielsen and Winskel. In each of these cases the added structure does not yield greater expressiveness: after a suitable unfolding the causalities expressed by this added structure are completely determined by the underlying transition system.

The families of configurations of (prime) event structures are given as sets of sets of events satisfying certain closure properties. The model of configuration structures is obtained by dropping all the closure conditions. ST-configuration structures are a further generalisation, in which the configurations may contain events 'partially' (in case they are currently being executed). Event automata, studied by Pinna and Poigné, fit between the configuration structures and the ST-configuration structures. Through appropriate translations these are shown to be equally expressive as so-called configuration-deterministic process graphs. Graphs which are not configuration-deterministic do not correspond to nets or event oriented models.

Further translations are provided between general P/T nets and ST-configuration structures, as well as the so-called higher dimensional automata. The ordinary configuration structures correspond with the Petri nets without self-loops. Propositional theories appear as a stepping stone in the translation from configuration structures to nets. These translations fit with the so-called collective token interpretation of Petri nets, as opposed to the individual token interpretation, which is needed to justify the unfolding of P/T nets into occurrence nets of Montanari, Messeguer and Sassone.

My translations are consistent with the correspondence between flow nets and flow event structures proposed by Boudol and Castellani, and with the correspondence between elementary net systems and elementary transition systems proposed by Rosenberg, Nielsen and Thiagarajan

Clemens Fischer, University of Oldenburg, Germany
Combining CSP and Z

The formal language Z is widely used for specifying a state space and operations on this state space. It is not designed for distributed systems where different processes act in parallel and communicate with each other. But exactly in this area lies the strength of the process algebra CSP which itself is not so well suited for specifying data and operations on these data. Nevertheless both aspects, communication and data, are important for the application of formal methods to industrial problems. In this work CSP and Z are combined to the new language CSP-Z. A CSP-Z specification consists of three parts: An interface, a CSP-process and Z-schemas. The interface declares the channels of the specification and assigns Z-types to every channel. The Z-schemas describe the local state space and operations on this state space for every channel from the interface. The semantic model and the definition of refinement is like CSP (possibly extended by states to deal with sequential composition). The semantic idea is that the Z-part and the CSP-process are executed in parallel. Hence, a trace is possible if it is possible for the CSP-process and if there is some internal state that is computed by sequential composition of the Z-schemas corresponding to the trace. A communication can be refused if it is refused by the CSP-process or the precondition of the corresponding Z-schema is false. Divergence is controlled only by the CSP-process.

With this approach refinement is preserved in the sense that refining the CSP-process results in a refinement of the CSP-Z specification. With some restriction this is also true for the Z concept of refinement. This makes it possible to use standard tools like the CSP model checker FDR.

The core of this idea together with a large number of transformation rules has already been investigated for the language MIX in the Esprit project ProCoS.

Ilaria Castellani, INRIA, France
Bisimulations for the asynchronous Π-calculus
(Joint work with Roberto Amadio and Davide Sangiorgi)

The asynchronous π-calculus is a variant of the π-calculus where message emission is non-blocking. Honda and Tokoro have studied a semantics for this calculus based on bisimulation. Their bisimulation relies on a modified transition system where, at any moment, a process can perform any input action.

In this talk we propose a new notion of bisimulation for the asynchronous π-calculus, based on the standard labelled transition system. We give several charcterizations of this equivalence, including one in terms of Honda and Tokoro's bisimulation and one in terms of barbed equivalence. We show that this bisimulation is preserved by name substitution and hence by input prefix. Finally we give a complete axiomatisation of the (strong) bisimulation for finite processes.

Ursula Goltz, University of Hildesheim, Germany
Causal Testing and Action Dependencies
(Joint work with Heike Wehrheim)

The talk consists of two parts:

Causal Testing [MFCS 96]:
An equivalence notion for event structures as a model of concurrent systems is suggested which combines the notion of testing (or failure) equivalence with respect to the timing of choices between different executions with a precise account of causalities between action occurrences as in causal semantics. This fills a gap in the lattice of equivalences considered in comparative concurrency semantics. It is shown that the new notion coincides with a "canonical" equivalence obtained as the usual testing performed on causal trees. Furthermore, it is shown that it is invariant under action refinement, thus fulfilling a standard criterion for non-interleaving equivalences.

Action Dependencies:
The approach of Mazurkiewicz trace theory - modelling causalities between action occurrences using a global dependency relation on actions - is generalised to branching time semantics (modelling both causalities and points of choice). More precisely, it is shown that the usual notion of bisimulation coincides with history preserving bisimulation and that usual testing coincides with causal testing for a system model with a global dependency relation.

Sven-Olof Nystron, Uppsala University, Sweden
No Fully Abstract Fixpoint Semantics for Non-Deterministic Languages With Infinite Computations

In a simple proof by contradiction I show that for a wide range of programming languages there cannot be a fully abstract least fixpoint semantics.

The proof applies to all mathematical structures in which functions have least fixed points.

Joost Kok, Leiden University, The Netherlands
Refinement for Coordination

We construct a framework for the development of programs based on coordination. Programs in this framework communicate through a shared dataspace. The structures are rather independent from the the programs. An example of a coordination language is the language Linda of Gelernter.

Compositionality plays an important role in coordination. A typical coordinated system has many, relatively independent components. It should be not too difficult to replace these components.

In order to construct such a framework, we use "semantic engineering". We combine three existing models for program development: UNITY, Action Systems and Gamma. We start from UNITY, add the locality and procedures of the Action Systems, and the tuple space constructs of Gamma.We base these additions on fully abstract semantic models.

We also show how the framework can be applied to the design of a small phone system.

Anders P. Ravn, Technical University of Denmark, Denmark
Events and States with Timers and Duration - A Design Exercise

We investigate a suitable design paradigm for a class of concurrent systems exemplified by a Brüel α Kjaer 2145 Vehicle Signal Analyzer. The characteristic features are: distributed sensing and preprocessing, and a computationally intensive process combining the time consistent measurements. The design proceeds through the following stages:

1) The interface is definded for each module by fixing an alphabet of channel-value pairs.
2) The communication behaviour is definded by one or more parallel processes in a suitable process algebra.
3) A local state is introduced for each process, and operations are defined for each communication linking state and communicated value.

4) If values from other processes or modules are needed, "shared" memories are introduced. Timing properties are in the design represented by symbolic tuner events. At the implementation level, it must be checked that the duration of operations and communications between consecutive timer events do not exceed the period of the hardware clocks. The main investigator in this work is Henrik Reif Andersen, who has demonstrated how CCS + TIMERS + ML are used in the design (TAPSOFT '95).

Michael Schenke, Oxford University, Great Britain
The Quickstep - Semantics of Handel

Handel is an OCCAM-like description language for Hardware-design. The actually existing compiler relies on a refinement calculus with transformation laws which are locking a formal, proof, transformation relation etc.

In the talk a model for a fairly expressive and manageable subset of this (even in its syntax) not clearly defined language is proposed, so that large parts of this calculus can be proven. The model is based on timing diagrams, programs are operations on them. From the tricky points of the semantics construction one has been chosen in order to illustrate the semantics alltogether, the treatment of timing. Some of the laws can be obtained in the model only due to a sequential operator which affects the past. The hardware construction proceeds by a development which starts with as much timing nondeterminism as possible. This nondeterminism is reduced by laws and at some stage by change of the semantics which invalidates some timed implementation laws but allows new algebraic transformation.

The interplay between parallelism with communication via links as in OCCAM and information exchange via shared variables leads to an implementation relation which makes the semantic domain into a preorder rather than into a cpo. The semantics of WHILE loops is defined by syntactic approximation.

Frank S. de Boer, Utrecht University, The Netherlands
Compositional State-Based Proof-Theories for Asynchronous Communication via Unbounded FIFO-Buffers.

A compositional reformulation of the non-compositional Apt-Francez-de Roever proof method for CSP is presented in an Invariant-logic for asynchronously communicating processes. Restricting to deterministic processes simplifies the logic in that auxiliary variable are not needed.

Moreover it is argued that a restriction to local non-determinism allows for a decomposition of the global Invariant. The latter proof method is applied to the correctness of an algorithm for computing the Network Topology in the interactive proof-checker PVS.

Ernst-Rüdiger Olderog, University of Oldenburg, Germany
Transformational Design of Real-Time Systems
(Joint work with Michael Schenke, Oxford University)

We present a transformational approach to the stepwise design of distributed real-time systems. The starting point of the design are global requirements formulated in a subset of Duration Calculus called "DC implementables" and the target are programs in an occam-like programming language PL. While Duration Calculus provides a state-based description of the real-time system, PL is event-based using the synchronous communication paradigm of occam. To bridge this gap we introduce an intermediate program specification language called SL. This language combines regular expressions with action systems and time conditions.

The transformational calculus relies on the "mixed term" approach in which pieces of DC implementable, SL and PL are mixed in a semantical coherent manner. An application of a transformation rule refines a given mixed term into a new mixed term describing the system on a more detailed level. Refinement is modelled here by logical implication in the Duration Calculus coupled with linking invariants to cater for changes of observable. The whole approach is illustrated by the example of a computer controlled gas burner.

Nicoletta Sabadini, University of Milano, Italy
An Automata Based Approach to Local Independence
(Joint work with RFC Walters, Univ. of Sydney)

We consider an automata-based model of distributed systems, "distributive automata". These automata are built in a compositional way from a given set of data types and data questions by using only operations allowable in a distributive category. The automata so obtained generalize Zielonka Automata by allowing an explicit representation of data, hierarchical descriptions, local independence. We show that recursive functions and BSS-recursive functions can be obtained. We provide a hybrid model for asynchronous circuits which allows the analysis both at the continuous and at the event level.

Volker Diekert, University Stuttgart, Germany

Rewriting Traces

(Joint work with Rance Cleaveland and V. Natarajan, N.C. State University)

Trace rewriting systems are an abstract model for transformations on concurrent processes. We focus on some basic algorithmic questions, in particular: how efficient irreducible normal forms can be computed.

We show an O(n log n) algorithm which applies in many cases and improves the O(n$^{2)}$ algorithm formerly known. The result has been presented first in a joint work with M. Bertol at STACS '96 in Grenoble.