05081 Abstracts Collection
Foundations of Global Computing
— Dagstuhl Seminar —

José Luiz Fiadeiro[1], Ugo Montanari[2] and Martin Wirsing[3]

[1] Univ. of Leicester, GB
jose@fiadeiro.org
[2] Univ. di Pisa, IT
ugo@di.unipi.it
[3] Univ. München, DE
wirsing@informatik.uni-muenchen.de

Abstract. From 20.02.05 to 25.02.05, the Dagstuhl Seminar 05081 on "Foundations of Global Computing" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Global computing

On declassification and the non-disclosure policy

Gérard Boudol (INRIA - Sophia Antipolis, F)

We address the issue of declassification in a language-based security approach. We introduce, in a Core ML-like language with concurrent threads, a declassification mechanism that takes the form of a local flow policy declaration. The computation in the scope of such a declaration is allowed to implement information flow according to the local policy. This dynamic view of information flow policies is supported by a concrete presentation of the security lattice, where the confidentiality levels are sets of principals, similar to access control lists. To take into account declassification, and more generally dynamic flow policies, we introduce a generalization of non-interference, that we call the non-disclosure policy, and we design a type and effect system for our language that enforces this policy.

Keywords: Confidentiality, language-based security, type and effect system

Joint work of: Boudol, Gérard; Almeida Matos, Ana

Dagstuhl Seminar Proceedings 05081
Foundations of Global Computing
http://drops.dagstuhl.de/opus/volltexte/2006/459

A process-algebraic approach to hybrid systems

Ed Brinksma (University of Twente, NL)

Process algebra is a theoretical framework for the modelling and analysis of the behaviour of concurrent discrete event systems that has been developed within computer science in past quarter century. It has generated a deeper understanding of the nature of concepts such as observable behaviour in the presence of nondeterminism, system composition by interconnection of concurrent component systems, and notions of behavioural equivalence of such systems. It has contributed fundamental concepts such as bisimulation, and has been successfully used in a wide range of problems and practical applications in concurrent systems.

We believe that the basic tenets of process algebra are highly compatible with the behavioural approach to dynamical systems. In our contribution we present an extension of classical process algebra that is suitable for the modelling and analysis of continuous and hybrid dynamical systems. It provides a natural framework for the concurrent composition of such systems, and can deal with nondeterministic behaviour that may arise from the occurrence of internal switching events. Standard process algebraic techniques lead to the characterization of the observable behaviour of such systems as equivalence classes under some suitably adapted notion of bisimulation.

Keywords: Process algebra, hybrid systems, control theory

Injecting distribution in CASL

Maura Cerioli (University of Genova, I)

We present a first attempt at the development of a library in the specification language Casl providing primitives to represent connectivity and communication in a distributed system.

The focus, in particular, is on peer-to-peer, which presents more challenges than the client-server paradigm, because of the higher degree of anarchy and the large amount of middleware providing similar but different features in support of it.

From our experience on the definition of this library, we draw some methodological lessons on how to deal with the capture of complex software systems, as opposite to classical libraries representing standard or mathematical datatypes.

Keywords: P2P, CASL, algebraic specification language, specification library

Joint work of: Cerioli, Maura; Dell'Amico, Matteo

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/298

Relative expressiveness of a stack of Klaim(s)

Rocco De Nicola (University of Firenze, I)

We shall discuss the expressive power of variants of Klaim, an experimental language with programming primitives for global computing that combines the process algebra approach with the coordination-oriented one. Klaim has proved to be suitable for programming a wide range of distributed applications with agents and code mobility, and has been implemented on the top of a runtime system written in Java. The expressivity of its constructs is tested by distilling from it a few, more and more foundational, calculi (namely, mu-Klaim, c-Klaim and lc-Klaim) and studying the encoding of each of the considered languages into a simpler one. The expressive power of Klaim based calculi is finally tested by comparing one of the calculi with the asynchronous pi-calculus. In particular, we describe and assess an encoding of c-Klaim in the asynchronous pi-calculus and an encoding of the asynchronous pi-calculus in lc-Klaim.

Joint work of: De Nicola, Rocco; Gorla, Daniele; Pugliese, Rosario

Uncovering interaction patterns in ad-hoc processes

Schahram Dustdar (TU Wien, A)

Global computing aims at providing a large-scale infrastructure for communications, computing, and collaboration between computational devices, resources, humans, and software (Web) services.

In this presentation I presented a software infrastructure which can be utilized for collaboration and coordination of human work activities and software (Web) services on the other hand.

Furthermore, mining techniques and metrics were discussed to uncover interaction patterns in such settings. In particular I presented the automatic generation of control flows visualized as Petri-Nets, activity-role diagrams, and sociograms visualizing collaboration and coordination.

Keywords: Ad hoc processes, web services, coordination, mining

A model checking framework for (Mobile) UML statecharts

Stefania Gnesi (CNR - Pisa, I)

In this paper we present an “on the fly” model checker for the verification of the dynamic behavior of UML models seen as a set of communicating state machines. The logic supported by the tool is the state/event-based temporal logic μUCTL that makes possible the description of properties on UML model evolutions and assertions on explicit local state variables of UML state machines.

This logic allows both to specify the basic properties that a state should satisfy, and to combine these basic predicates with advanced logic or temporal operators. Doubly Labelled Transition Systems are the semantic domain for μUCTL where states are labelled by sets of propositions that hold in them and transitions by events performed.

The logic we propose here is then applied to verify properties over the dynamic behaviour of a mobile system modelled as extended UML statechart.

Keywords: Mobile UML statecharts, model checking, action/state based temporal logic

Joint work of: Gnesi, Stefania; Mazzanti, Franco

Basic observables for a calculus for global computing

Daniele Gorla (Università di Roma ”La Sapienza”, I)

We introduce a foundational language for modelling applications over global computers whose interconnection structure can be explicitly manipulated. Together with process distribution, mobility, remote operations and asynchronous communication through distributed data spaces, the language provides constructs for explicitly modelling inter-node connections and for dynamically establishing and removing them. For the proposed language, we define natural notions of extensional observations and study their closure under operational reductions and/or language contexts to obtain barbed congruence and may testing equivalence. For these equivalences, we provide alternative characterizations in terms of labelled bisimulation and traces that can be used for actual proofs. Finally, we briefly sketch an application of the language (and of its semantic theories) to a non-trivial routing scenario that can be equationally proved sound.

Keywords: Observational equivalences, net topology, process calculi

Joint work of: De Nicola, Rocco; Gorla, Daniele; Pugliese, Rosario

Data Handover: reconciling message passing and shared memory

Jens Gustedt (INRIA Lorraine, F)

Data Handover (DHO) is a programming paradigm and interface that aims to handle data between parallel or distributed processes that mixes aspects of message passing and shared memory.

It is designed to overcome the potential problems in terms of efficiency of both:

1. memory blowup and forced copies for message passing and
2. data consistency and latency problems for shared memory.

Our approach attempts to be simple and easy to understand. It contents itself with just a handful of functions to cover the main aspects of coarse grained inter-operation upon data.

Keywords: Efficient data management, message passing, shared memory

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/297

A theory of system behaviour in the presence of failures

Matthew Hennessy (University of Sussex, GB)

We develop a behavioural theory of distributed systems in the presence of failures. The framework we use is that of Dpi, a language in which located processes, or agents, may migrate between dynamically created locations. These processes run on a distributed network, in which individual nodes may fail, or the links between them may be broken. The language is extended by a new construct for detecting, and reacting to these failures.

We define a bisimulation equivalence between these systems, based on labelled actions which record, in addition to the effect actions have on the processes, the actual state of the underlying network and the view of this state known to observers. We prove that the equivalence is fully abstract, in the sense that two systems will be differentiated if and only if, in some sense, there is a computational context, consisting of a network and an observer, which can see the difference.

Keywords: Distributed systems, failures, pi-calculus, behavioural equivalence, bisimulation equivalence, full abstraction

Joint work of: Hennessy, Matthew; Francalanza, Adrian

The impact of class transformation patterns on State Machines

Piotr Kosiuczenko (University of Leicester, GB)

UML offers a variety of diagrams for modelling various aspects of software. Most of UML diagrams are well understood in isolation, but their mutual relationship is much less understood. With few exceptions, there is even less understanding of the impact changes that one kind of diagram has on another kind of diagram. In particular, it was not clear which properties are preserved and what is the impact on other UML diagrams such as State Machines. In this talk we will discuss the impact of transformation patterns on class constraints, in particular OCL constraints. We will investigate the impact of class transformation patterns on State Machines. States in a State Machine are interpreted by state invariants, therefore structural relations between them can be interpreted as logical relations between the corresponding formulas. Preservation of the entailment relation can be seen as preservation of State Machine structure. We will present a sufficient condition for class structure transformation to preserve State Machine structure.

Keywords: UML, State Charts, redesign, refactoring, design patterns, OCL

Towards mobile distributed machines

Thomas A. Kuhn (TU München, D)

The author presents his ongoing PhD work towards a Mobile Distributed Machine Definition for (secure) mobile systems. The presentation contains two main parts: one describing an approach from earlier work of the author of mobility extensions of Interacting State Machines with an application to a simple example with mobile code. We introduce Ambient ISMs whose features besides others include hierarchical environments, migration, and locality constraints on communication. The second part of the talk contains a position statement of ongoing and further Phd work like issues of generalization, composition, refinement and application.

Keywords: Mobile systems

Insights emerged while comparing three models for global computing

Ivan Lanese (Università di Pisa, I)

In this paper we outline the main ideas emerged while studying a chain of mappings from *Fusion calculus* to *logic programming*, using *Synchronized Hyperedge Replacement* (with both Hoare and Milner synchronizations) as intermediate step. We aim more at discussing the ideas behind the mappings than at presenting their technical details.

Keywords: Fusion calculus, graph transformation, Synchronized Hyperedge Replacement, logic programming, mobility

Joint work of: Lanese, Ivan; Montanari, Ugo

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/295

Mapping Fusion and Synchronized Hyperedge Replacement into Logic Programming

Ivan Lanese (Università di Pisa, I)

We compare three different formalisms that can be used in the area of models for distributed, concurrent and mobile systems. In particular we analyze the relationships between a process calculus, the *Fusion calculus*, graph transformations in the *Synchronized Hyperedge Replacement* with Hoare synchronization (HSHR) approach and *logic programming*. We present a translation from Fusion calculus into HSHR (whereas Fusion calculus uses Milner synchronization) and prove a correspondence between the reduction semantics of Fusion calculus and HSHR transitions. We also present a mapping from HSHR into a transactional version of logic programming and prove that there is a full correspondence between the two formalisms. The resulting mapping from Fusion calculus to logic programming is interesting since it shows the tight analogies between the two formalisms, in particular for handling name generation and mobility. The intermediate step in terms of HSHR is convenient since graph transformations allow for multiple, remote synchronizations, as required by Fusion calculus semantics.

Keywords: Fusion calculus, graph transformation, Synchronized Hyperedge Replacement, logic programming, mobility

Joint work of: Lanese, Ivan; Montanari, Ugo

Context-awareness in software architectures

Antonia Lopes (University of Lisboa, P)

Much of the work in context-aware computing has been devoted to the development of technical solutions. In this talk, we aim for higher levels of abstraction and address the integration of context-awareness in the set of aspects that Software Architectures should be able to deal with.

More concretely, we show how the description of context-awareness aspects of systems can be integrated with the techniques that we have been developing within the IST-2001-32747 project AGILE on Architectures for Mobility for supporting distribution and mobility in Software Architectures. We propose an approach that allows software architects (i) to represent and organise the contextual information that a system requires and (ii) to take advantage of contextual information in the specification of the components and connectors of the system. The approach is based on the extension of software architectures with a new design element – context models.

By awarding a first-class status to the notion of context, the approach promotes the separation of concerns: it supports the description of context dependencies of a system's architecture in an explicit way through context models that can be understood independently of the specification of the system behaviour; it allows these aspects to be refined and evolved independently of the other concerns.

We illustrate the approach around a image search system.

Joint work of: Lopes, Antonia; Fiadeiro, José Luiz

Implementing mobile calculi

Michele Loreti (University of Firenze, I)

We shall present IMC, a Java software framework for building infrastructures to support the development of applications for systems where mobility and network awareness are key issues. The framework is particularly useful to develop runtime support for languages oriented towards global computing. The key features of the framework are illustrated by discussing the experience of implementing Dpi.

Specification and refinement of mobile computation in MTLA

Stephan Merz (INRIA Lorraine & LORIA - Nancy, F)

We present the spatio-temporal logic MTLA, an extension of Lamport's Temporal Logic of Actions TLA intended for the specification, verification, and formal development of systems including mobile code. MTLA is interpreted over ω-sequences of configurations, represented as finite trees of nested locations, endowed with local state. The logic contains a spatial modality to refer to sublocations; transitions can modify the local states, the topological structure, or both. We illustrate the formalism at the hand of a simple example of a shopping agent. We also present notions of refinement for mobile systems, and show that refinement can be represented as (validity of) implication, possibly after hiding local state components.

Keywords: Mobile code, specification, verification, refinement, temporal logic, spatial logic, TLA

Joint work of: Merz, Stephan; Wirsing, Martin; Zappe, Julia

Simplification and computation

Eugenio Moggi (University of Genova, I)

We present a framework for operational semantics related to the Chemical Abstract Machine and the operational semantics for monadic metalanguages. It involves two relations: simplification (a confluent relation on terms) and computation (based on multiset rewriting). We consider specific instances, with expressive pattern-matching facilities and join patterns, which distinguish atoms from variables (as in FreshML).

Keywords: Operational semantics, term rewriting, multiset rewriting

Incremental proof-based development - SoC

Dominique Méry (LORIA - Nancy, F)

Formal methods provide techniques and tools for constructing mathematical models of software systems; we develop mathematical models of a monitoring tool for measurement in Digital Video Broadcasting Television (DVB-T). The case study shows the way to develop formal models and it shows that interactions with non-specialists partners are possible. Moreover, graphs over parameters related to the monitoring process, and called dependency graphs, are derived from abstract mathematical models of the system; they express a hierarchy among

parameters, validated by proof of invariance and preserved by refinement; the hierarchy is incrementally built and provides hints for the future architecture of the SoC: decisions of implementations for computing parameters.

Finally, the refinement process expresses a relationship over system models and over dependency graphs. Dependency graphs derived from invariant, provide a hint for organizing the computation of parameters on a chip, since costs and size are important indicators for producing chips. Our methodology is based on the B event-based method, which integrates the incremental development of models using a theorem prover to validate each step of development called refinement. To complete the process, we analyse the translation of models into SystemC programs and we model the scheduler of SystemC to validate the translation of formal models into SystemC programs.

Keywords: Formal method, B event-based method, refinement, safety, architecture, terrestrial television

Unreliable failure detectors via operational semantics

Uwe Nestmann (EPFL - Lausanne, CH)

The concept of unreliable failure detectors for reliable distributed systems was introduced by Chandra and Toueg as a fine-grained means to add weak forms of synchrony into asynchronous systems. Various kinds of such failure detectors have been identified as each being the weakest to solve some specific distributed programming problem.

In this paper, we provide a fresh look at failure detectors from the point of view of programming languages, more precisely using the formal tool of operational semantics.

Inspired by this, we propose a new failure detector model that we consider easier to understand, easier to work with and more natural.

Using operational semantics, we prove formally that representations of failure detectors in the new model are equivalent to their original representations within the model used by Chandra and Toueg.

Keywords: Distributed algorithms, failure detectors, operational semantics

Joint work of: Nestmann, Uwe; Fuzzati, Rachele

Architectural views for CommUnity

Cristóvão Oliveira (University of Leicester, GB)

CommUnity and its categorical foundations provide a formal approach to Software Architecture (SA). Several concepts such as (re) configuration and (higher-order) connectors have been given precise definitions in this setting.

One of the cornerstones of the approach is the separation between computation, coordination and distribution. In this paper, we take this separation one step further and define explicit architectural views, one for each concern. They will be used to help to detect errors made while building the architecture. Moreover they will be a support to improve the design of the system by focusing on one concern at a time and/or by combining them with each other.

Keywords: Software architecture, views, computation, coordination, distribution

Joint work of: Oliveira, Cristóvão; Wermelinger, Michel

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2006/296

Probabilistic anonymity

Catuscia Palamidessi (INRIA Rhône-Alpes, F)

The concept of anonymity comes into play in a wide range of situations, varying from voting and anonymous donations to postings on bulletin boards and sending mails. A formal definition of this concept has been given in literature in terms of nondeterminism. In this paper, we investigate a notion of anonymity based on probability theory, and we we discuss the relation with the nondeterministic one. We then formulate this definition in terms of observables for processes in the probabilistic π-calculus, and propose a method to verify automatically the anonymity property. We illustrate the method by using the example of the dining cryptographers.

Keywords: Anonymity, probability theory, process calculi

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/299

Self-Service: An architecture for service composition

Stephan Reiff-Marganiec (University of Leicester, GB)

In this talk we examine a proposed architecture that allows automatic assembly of services by end-users. This is emerging work.

We discuss how the architecture

- can support static service composition by use of flow languages,
- could support fully automatic composition by using semantic web research activities and
- (my own current interest) how it can be used to assemble services based on user policies.

In particular the latter aspect will be supported by such an architecture being in place in the telecommunications domain.

Keywords: Automatic service composition, architecture

Analysis of an electronic voting protocol in the applied pi-calculus

Mark D. Ryan (University of Birmingham, GB)

Keywords: Verification, applied pi-calculus, electronic voting, secure protocol

Joint work of: Ryan, Mark D.; Kremer, Steve

BiLog: A structural (nominal) logic

Vladimiro Sassone (University of Sussex, GB)

Bigraphs are emerging as an interesting model for concurrent calculi, like CCS, pi-calculus, and Petri nets. Bigraphs are built orthogonally on two structures: a hierarchical place graph for locations and a link (hyper-)graph for connections. With the aim of describing bigraphical structures, we introduce a general framework for logics whose terms represent arrows in monoidal categories. We then instantiate the framework to bigraphical structures and obtain a logic that is a natural composition of a place graph logic and a link graph logic.

We explore the concepts of separation and sharing in these logics and we prove that they generalise some known spatial logics for trees, graphs and tree contexts.

Joint work of: Sassone, Vladimiro; Conforti, Giovanni; Macedonio, Damiano

Practical techniques for language design and prototyping

Mark-Oliver Stehr (Univ. of Illinois - Urbana, USA)

Global computing involves the interplay of a vast variety of languages, but practically useful foundations for language specification and prototyping at the semantic level are lacking.

In this talk we present a systematic approach consisting of three techniques:

1. A generic calculus of explicit substitutions with names (called CINNI) that allows us give a first-order representation of syntax to uniformly deal with all binding aspects.
2. An executable representation of Felleisen-style operational semantics in terms of first-order rewrite rules.
3. A logical framework, namely rewriting logic, that allows us to express (1) and (2) and, in addition, language aspects such as concurrency and nondeterminism.

We illustrate the use of these techniques in two applications:

1. A formal specification and analysis of PLAN, a Packet Language for Active Networks, that has been developed in the Switchware project at UPenn. This work was conducted in the scope of the DARPA Active Network Program.
2. The development of CIAO, a Calculus of Imperative Active Objects, a core language for concurrent object-oriented programming. It is especially designed to allow the representation of practically relevant sublanguages of common object-oriented languages such as Java, C#, and C++. This second application is subject of ongoing work.

Keywords: Rewriting logic, explicit substitutions, operational semantics, active networks, active objects

Joint work of: Stehr, Mark-Oliver; Talcott L., Carolyn

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/300

Distributed components and the Kell calculus

Jean-Bernard Stéfani (INRIA Rhône-Alpes, F)

Distributed systems programming is moving towards component-based models, including for the construction of middleware layers themselves. The talk will present an example reflective component-model called Fractal, which has been used in the construction of asynchronous middleware and communication subsystems. It will then discuss the formalization of the Fractal operational semantics via an interpretation of the Fractal constructs in the Kell calculus, a higher-order process calculus with localities. The talk will conclude with a presentation of recent results on Kell calculus equivalences and a discussion of further research directions.

Keywords: Component-based programming, distributed process calculi, higher-order process calculi

Joint work of: Stéfani, Jean-Bernard; Schmitt, Alan

MiKO—Mikado Koncurrent Objects; An instance of the MIKADO migration model

Vasco T. Vasconcelos (University of Lisboa, P)

The motivation for the Mikado migration model is to provide programming constructs for controlling code mobility that are as independent as possible from the particular programming language used to program the code. The main idea is to regard a domain (or site, or locality), where mobile code may enter or exit, as a membrane enclosing running processes, and offering services that have to be called for entering or exiting the domain.

MiKO—Mikado Koncurrent Objects is a particular instance of this model, where the membrane is explicitly split in two parts: the methods defining the interface, and a process part describing the data for, and the behavior of, the interface.

The talk presents the syntax, operational semantics, and type system of MiKO, together with an example. It concludes by briefly mentioning the implementation of a language based on the calculus.

Keywords: Global computing, code migration, administrative domains, process calculus

Joint work of: Martins, Francisco; Salvador, Liliana; Lopes, Luís; Vasconcelos, Vasco T.

Full Paper: http://drops.dagstuhl.de/opus/volltexte/2006/301

Modules and contracts for pi-calculus

Lucian Wischik (Microsoft Corp. - Redmond, USA)

We add two statements to pi, ”import” and ”export”, which are enough to model interfaces and modules (such as web-services, or local services, or just separate units of compilation on a single computer). The interfaces amount to behavioural contracts for modules. They allow us to guarantee global properties (such as deadlock-freedom) with purely local compile-time checks.

Keywords: Pi-calculus, modules, behavioural types, contracts, web services, type checking

Declarative synchronization for reconfigurable, communicating systems

Pawel T. Wojciechowski (EPFL - Lausanne, CH)

We present a language of concurrency combinators that allows a program code and its synchronization policy to be expressed separately. The language allows policies to be declared in programs, instead of having to be encoded using the low-level synchronization constructs.

The policies include: true parallelism, sequentiality, and isolation (or serializability). The language is equipped with a type system. The type system is used to verify if a policy declared using combinators can be satisfied by program execution.

In the second part, we focus on the isolation policy and present a language and runtime support of isolation-only transactions (called tasks). Tasks may have irrevocable I/O effects. The key concept of our design is the use of a type system to support rollback-free and safe runtime execution of tasks.

We present versioning concurrency control algorithms, and summarize a first-order type system that we have designed to verify information for the Basic Versioning algorithm.

Finally, we sketch the proof of type soundness.

Keywords: Programming languages, concurrency, type theory, transactions, isolation, concurrency control algorithms, declarative synchronization, abstract types, lambda calculus

An institution for mobile components

Artur Zawlocki (University of Warsaw, PL)

We present a formalism for the specification of systems made of interacting components. A model of such a system consists of a number of labelled transition systems, modelling individual components, and partial morphisms between them, representing relationships between those components.

Partiality of morphisms allows us to model component reconfiguration. In particular, when some of the components are interpreted as locations, reconfiguration allows us to capture mobility as changes of distributed system structure.

The formalism forms an institution. The semantic part is accompanied by logic which can express properties of individual components, as well as of systems built by interconnecting those components. Having an institution allows us to apply formalisms such as structured and architectural specifications in the domain of distributed systems.