

*International Workshop on*  
**FORMAL METHODS and SECURITY**

( IWFMS 2004 )

**May 17 - 20, 2004**  
**Nanjing, P. R. China**

This workshop aims at highlighting the importance of formal tools for the design and implementation of programming languages and systems, both conventionally sequential and concurrent. It arises as the joint initiative of the Department of Computer Science in Nanjing University and of the laboratory Preuves, Programmes et Systèmes (CNRS and Université Paris 7). Local organization by the Department of Computer Science in Nanjing University and the State Key Laboratory of Novel Software Technology (Nanjing).

**Program Committee**

Guy COUSINEAU	(Université Paris 7)
Pierre-Louis CURIEN	(CNRS/Université Paris 7, co-chair)
Stéphane GRUMBACH	(INRIA)
Jian LÜ	(State Key Laboratory, Nanjing University)
Fang-Min SONG	(Nanjing University, co-chair)
Ying JIANG	(Chinese Academy of Sciences)

**Local Organizing Committee**

Jian LÜ	(State Key Laboratory, Nanjing University)
Fang-Min SONG	(State Key Laboratory, Nanjing University)
Xiang-Lin FEI	(State Key Laboratory, Nanjing University)

## Schedule

### Monday (May 17)

- 09:00-09:30 Registration
- 09:30-09:40 Opening of IWFMS'04
- 09:40-10:30 Invited Talk<sup>1</sup>: Gilles Dowek. *Modeling and Verification of an Air Traffic Concept of Operation.*
- 10:30-10:40 Break
- 10:40-11:00 Jing Chen. *Local model checking real-time value-passing system.*
- 11:00-11:20 Wan Fokkink, Jun Pang. *Cones and foci for timed transition systems verification revisited*
- 11:20-11:40 Coffee Break
- 11:40-12:30 Invited talk: David Nowak. *Logical Relations for Monadic Types.*
- 12:30-14:30 Lunch Break
- 14:30-15:20 Invited talk: Jean-Jacques Lévy. *Confluent calculi for history-based control flow analysis.*
- 15:20-15:30 Break
- 15:30-15:50 Chao Qin, Zhong Chen. *Using combined method to analyze security protocols.*
- 15:50-16:10 Yu Zhang. *Extending logical relations for dynamic name creation with encryption.*
- 16:10-16:30 Coffee Break
- 16:30-17:20 Invited Talk: James J. Leifer. *Language design for distributed computing: Abstraction, Rebinding and Version Control.*
- 17:20-17:30 Break
- 17:30-18:30 Discussion Time

### Tuesday (May 18)

- 09:30-10:20 Invited Talk: Giuseppe Castagna. *Toward a general definition of non-interference for mobile systems.*
- 10:20-10:30 Break

---

<sup>1</sup>An invited talk is 50 minutes long, including 10 minutes for questions. A contribution talk is 20 minutes long, including 5 minutes for questions.

- 10:30-11:20 Invited Talk: Guo-Qiang Zhang. *Theory and Applications of Formal Concept Analysis.*
- 11:20-11:40 Coffee Break
- 11:40-12:00 Sylvain Baro, François Maurel. *The  $qv$  and  $qvK$  calculi: name capture and control.*
- 12:00-12:20 Emmanuel Beffara. *Realizability with constants.*
- 12:20-12:40 Xiao-Cong Zhou. *Insertion and categorical model of higher-order subtyping.*
- 12:40-14:30 Lunch Break
- 14:30-14:50 Qin Ma, Luc Maranget. *Compiling pattern matching in join-patterns.*
- 14:50-15:10 Vicent Simonet. *The flow Caml system: information flow analysis in practice.*
- 15:10-15:20 Break
- 15:20-16:10 Invited Talk: Jean-François Monin. *Formalisation of the Join-calculus in Coq.*
- 16:10-16:40 Coffee break
- 16:40-17:00 Shin-ya Nishizaki. *Secure execution of client-side scripts by program transformation in an HTTP proxy server.*
- 17:00-17:20 Hui Xu, Ai-Min Pan. *Formal modeling of event correlation in IDS.*
- 17:20-17:30 Break
- 17:30-18:30 Invited Talk: Pierre Crégut. *Java on Mobile Phones: A New Playground for Formal Methods.*

### Wednesday (May 19)

- 09:30-10:20 Invited Talk: Jean-Louis Lanet. *The use of Formal Methods in the Smart Card Industry (an experience report).*
- 10:20-10:30 Break
- 10:30-10:50 Jens Chr. Godskesen, Thomas Hildebrandt. *Copyability types for mobile computing resources.*
- 10:50-11:10 Yu-Xin Deng. *On mobility and communication.*
- 11:10-11:30 Coffee Break

- 11:30-12:20 Invited Talk: Hui-Min Lin. *A Predicate Spatial Logic for Mobile Processes.*
- 12:20-14:30 Lunch Break
- 14:30-14:50 Pascal Cuoq, Sunae Seo, Hongseok Yang, Kwangkeun Yi. *Proof compilation.*
- 14:50-15:10 Tao-Lue Chen, Yang-Yue Feng, Jian Lü. *Extension of type evolution system for robust mobile ambient.*
- 15:10-15:20 Break
- 15:20-16:10 Invited Talk: Ji-Feng He. *Linking theories of concurrency*
- 16:10-16:30 Coffee break
- 16:30-17:20 Invited Talk: Yu-Xi Fu. *On open equivalence.*
- 17:20-18:20 Free Time (Discussion)
- 18:20-18:30 Closure

#### **Thursday (May 20)**

This day is arranged for an one-day excursion.

## Invited Talks

### Giuseppe CASTAGNA

*Toward A General Definition of Non-interference for Mobile Systems*

(joint work with Samuel Hym, Michele Bugliesi and Sabina Rossi).

The aim of this work is to state a general definition of non-interference for concurrent and mobile calculi. By general we mean that it can be applied to different mobility frameworks by very slight modifications. As a first attempt we apply it to CCS and the pi-calculus and devise types systems to statically check it.

### Pierre CREGUT

*Java on Mobile Phones: A New Playground for Formal Methods.*

Most modern mobile phones contain a Java runtime environment that can execute programs downloaded by the customer. Opening the phone with downloadable programs creates new security risks for the customer and even if Java provides intrinsic security mechanisms, the assets of the customer could be jeopardized if any syntactically valid program was accepted on the phone.

Because Java for mobile environment (J2ME) is small but still very powerful, customizable but still relatively closed, it is an interesting playground for formal methods. Properties to check range from simple security properties of programs that must be checked automatically to complex properties of the platforms that require more sophisticated verification procedures.

In this talk, we will insist on the differences with the verification of Java-Card programs, an active research area during the last years, we will present some preliminary results on the verification of “midlets” (J2ME programs for mobile phones using the MIDP configuration) based on the use of static analysis techniques and we will provide hints on new opened challenges and new research directions.

### Gilles DOWEK

*Modeling and Verification of an Air Traffic Concept of Operation*

(joint work with César Muñoz and Victor Carreño)

In this talk we will describe an air traffic concept of operation for small airports will little facilities. I will discuss various techniques (model checking, theorem proving, ...) that can be used to establish some properties of this concept of operations.

### Yu-Xi FU

*On Open Equivalence.*

By focusing on a simple calculus of nondeterministic mobile processes, it is shown that Milner’s three tau laws fail to lift a complete system for strong open congruence to a complete system for weak open congruence in the presence of match operator. A fourth tau law is proposed that deals with match operator

under prefix operation. It is shown that a complete system for the strong open congruence extended with all the four tau laws is complete for the weak open congruence. These observations have shed light on later research.

## **Ji-Feng HE**

*Linking Theories of Concurrency.*

A fundamental concept of any theory of concurrency is an ordering relation, which enables similar concurrent processes to be compared in some meaningful way. Different theories distinguish themselves in their choice of definition of this ordering. Commonly, the ordering is symmetric (an equivalence), and is known as bisimilarity. The strongest of the asymmetric definitions is strong simulation, and the weakest one is the trace inclusion. Ordering based on simulation are conducive to simple proofs by induction, and the inequalities are susceptible to mechanical verification by method checking. Orderings based on refinement are useful in program specification, design and optimization.

This work aims to combine the merits of both kinds of ordering: the simple proofs offered by simulation and wide applicability of refinement. We follow the approach advocated in the book “Unifying theories of programming” by exploring a collection of healthiness conditions and introducing a link function  $H$ . For theories of concurrency, the healthiness conditions will be expressed as extra transition rules, and added to the operational semantics of any process calculus, independent of the details of its particular syntax and operational semantics. The pair  $(Id, H)$ , as a Galois connection, is going to link theories of concurrency based on simulation with those based on refinement.

## **Jean-Louis LANET**

*The use of Formal Methods in the Smart Card Industry (an experience report).*

Smart cards could be the ideal domain for applying formal methods for mainly three reasons : mastering the complexity of the new operating systems, certifying at a high level a part of the smart card and reducing the cost of the validation. We believed that these reasons were enough to introduce formal methods in the software live cycle. Recently an original framework developed within the CASSIS team at INRIA for generating test cases from formal models has been successfully applied for smart card applications. Other solutions for low level validation include the JML framework and tools developed with EVEREST (Jack) and LOGICAL (Krakatoa). For the certification, the high level certification (E-6 in the ITSEC framework) obtained by Multos in 1999 did not encourage the other smart card manufacturers to propose such high level certification due to the costs, even if several manufacturers got EAL5+ certificates. If certification helps to introduce formal methods it is just as a side effect. Finally it is the complexity of the operating systems and the need to avoid vulnerabilities that initiated some smart card developments at Gemplus.

After an introduction to different initiatives to apply formal methods in the smart card industry we will present a practical experience of using formal methods for development and a trial to compare a conventional development and a

formal on. The correct design and implementation of a smart card is the key to shun a logical attack that try to exploit code vulnerabilities. Open smart cards like Java Card provide application developers an opportunity to develop rapidly applications by offering the possibility to download during post issuance application into the card. The main drawback with this kind of smart cards is the risk to download a hostile application that may exploit a faulty implementation module of the platform. Security is always a big concern for smart cards, but the issue is getting more intense with multi-applicative platforms, post issuance code downloading and the constant growth of the complexity. The card becomes an open system with communicating applications sharing the same resources. Integrating a Java byte code verifier into the card is the first step to ensure the dependability and the integrity of the card allowing to ensure its own security. For that purpose such a module need to be developed with formal techniques.

The main drawback with formal methods is the lack of publicly accessible data about the overhead in the development process and the cost of using formal methods. We believe that a clean methodology with related metrics and tools improvements will consequently help the integration of formal methods in the software process.

### **James J. LEIFER**

*Language Design for Distributed Computing: Abstraction, Rebinding and Version Control*

(joint work with Gilles Peskine, Peter Sewell and Keith Wansbrough).

We discuss the design of programming languages for distributed computation, focusing on support for type-safe marshalling of arbitrary language values. In particular: (1) unmarshalling can involve rebinding to local resources; (2) values of abstract types can be communicated, and a globally-coherent notion of type equality ensures that unmarshalling respects abstraction; and (3) interoperation between separately-built programs with different versions of shared modules is supported, with fine-grain version control. This paper discusses the design space and describes an experimental language, Acute, which collects a coherent set of design choices. Acute extends an ML fragment with marshalling and versions, it has a complete semantic definition (of typing, compilation, and runtime), and has been implemented in FreshOcaml.

### **Jean-Jacques LÉVY**

*Confluent calculi for history-based control flow analysis*

(joint work with Tomasz Blanc).

A calculus for Stack Inspection has been used by Fournet and Gordon to model the behavior of security managers implemented in runtimes such as the JVM and the CLR. Based on this model, a static analysis technique has been applied to *C#* libraries (work by Blanc, Fournet, Gordon). Abadi and Fournet have proposed informal methods for access control based on execution history. In this talk, we consider a confluent calculus for this kind of control flow analysis.

It is based on a confluent calculus of lambda expressions with history (using so-called Levy's labels). We claim that confluency is an important property for the design of static analysers. (This talk mainly covers work in progress)

### **Hui-Min LIN**

*A Predicate Spatial Logic for Mobile Processes.*

A modal logic for describing temporal as well as spatial properties of mobile processes, expressed in the asynchronous  $\pi$ -calculus, is presented. The logic has recursive constructs built upon predicate-variables. The semantics of the logic is established and shown to be monotonic, thus guarantees the existence of fixpoints. An algorithm is developed to automatically check if a mobile process has properties described as formulas in the logic. The correctness of the algorithm is proved.

### **Jean-François MONIN**

*Formalisation of the Join-calculus in Coq.*

Towards verifying mobile code, we choose to formalize an appropriate elementary model of concurrency in type theory (more precisely: the calculus of inductive constructions, CIC, implemented in the Coq proof assistant).

The chosen process calculus is the Join-Calculus (JC) of Fournet, Conthier and Levy. The latter can be seen as a variant of the pi-calculus of Milner, Parrow and Walker, with better locality properties. We use Coq for encoding a structural operational semantics of JC based on the reflexive chemical abstract machines model. Our formalization uses the full power of inductive constructions and dependent types available in the underlying theory of Coq, in order to capture the reflexive features of JC.

### **David NOWAK**

*Logical Relations for Monadic Types.*

Logical relations and their generalizations are a fundamental tool in proving properties of lambda-calculi, e.g., yielding sound principles for observational equivalence. We propose a natural notion of logical relations able to deal with the monadic types of Moggi's computational lambda-calculus. The treatment is categorical, and is based on notions of subconing and distributivity laws for monads. Our approach has a number of interesting applications, including cases for lambda-calculi with non-determinism (where being in logical relation means being bisimilar), dynamic name creation, and probabilistic systems.

### **Guo-Qiang ZHANG**

*Theory and Applications of Formal Concept Analysis.*

Formal concept analysis (FCA) is a powerful lattice-based tool for symbolic data analysis and ontological engineering. We propose a new approach to formal concept analysis based on ideas motivated from domain theory, a subject of extensive study in theoretical computer science and programming languages.

The domain-theoretic idea of partial information and successive approximation suggests that for infinite structures to be computationally feasible, items of knowledge or information should either be finitely representable or approximable. A new notion called approximating concepts is introduced. The corresponding approximating concept lattices associated with (infinite) contexts are exactly the complete algebraic ones and every (classical) formal concept is approximating. Furthermore, in the case that the formal context is finite, approximating concepts and formal concepts coincide. We also introduce an appropriate notion of morphisms on formal contexts and show that the resulting category is equivalent to

(a) the category of complete algebraic lattices and Scott continuous functions and

(b) a category of information systems and approximable mappings, and hence is cartesian closed. Traditionally FCA has focused on finite structures; infinite contexts and their corresponding lattices are of theoretical and practical interest since they may offer connections with and insights from other mathematical structures which are normally not restricted to the finite cases. Applications in web-menu design as well as security policies will be discussed. This talk is based on joint work with Pascal Hitzler and Gongqin Shen.

## Accepted Papers

- [1] Sylvain Baro, François Maurel. *The  $qv$  and  $qvK$  calculi: name capture and control.*
- [2] Emmanuel Beffara. *Realizability with constants.*
- [3] Jing Chen. *Local Model Checking Real-time Value-passing System.*
- [4] Tao-Lue Chen, Yang-Yue Feng, Jian Lü. *Extension of Type Evolution System for Robust Mobile Ambient.*
- [5] Jens Chr. Godskesen, Thomas Hildebrandt. *Copyability Types for Mobile Computing Resources.*
- [6] Pascal Cuoq, Sunae Seo, Hongseok Yang, Kwangkeun Yi. *Proof Compilation.*
- [7] Yu-Xin Deng. *On Mobility and Communication.*
- [8] Wan Fokkink, Jun Pang. *Cones and Foci for Timed Transition Systems Verification Revisited.*
- [9] Qin Ma, Luc Maranget. *Compiling Pattern Matching in Join-Patterns.*
- [10] Shin-ya Nishizaki. *Secure Execution of Client-Side Scripts by Program Transformation in an HTTP Proxy Server.*
- [11] Chao Qin, Zhong Chen. *Using Combined Method to Analyze Security Protocols.*
- [12] Vicent Simonet. *The flow Caml system: information flow analysis in practice.*
- [13] Hui Xu, Ai-Min Pan. *Formal Modeling of Event Correlation in IDS.*
- [14] Yu Zhang. *Merging Encryption into Kripke Logical Relations of Dynamic Name Creation.*
- [15] Xiao-Cong Zhou. *Inserter and Categorical Model of Higher-order Subtyping.*