Esprit WG 24512

COORDINATION

—

From Coordination Models to Applications

Final Report

97/08/15 – 2000/11/14

Background

The roots of the Coordina WG (Working Group) are to be found in the Esprit Basic Research Project 9102 "Coordination".

This project allowed the fostering of a European research community around the theme of Coordination, understood as the principle of separation of concerns, in software development, between internal functionality of components and their coordinated interactive behaviour.

By the end of that project the Coordination community had grown larger, a first international conference was held, and it was felt that a Working Group would be a suitable format to keep the larger community interacting.

The Working Group

Duration

The Coordina WG was approved for the duration of three years, starting on August 15, 1997.

In the third year a three month extension was requested and granted, so the period of activity was from August 15, 1997 to November 14, 2000.

Objectives

The motivation behind the Coordination idea is not going to go away soon. Software systems are growing in size and complexity, concurrency and distribution are ever more present, and the complexity of understanding, developing and maintaining such systems is a real concern, entailing ever higher costs and longer deployment times.

Coordination methodologies propose to tackle these problems by promoting the separation of local functional concerns from global coordination aspects, at the various levels of software development, leading to the explicit specification of coordination fragments that can be better analysed and reused.

The main objective of the WG was to establish a European leadership in such a critical field, by joining theoreticians and practitioners in a collaborative assessment of diverse approaches, with a view towards consolidating the foundational work and identifying promising technological avenues worth exploring.

Planned activities

The group proposed to attain its goals through the following activities.

Foundational work

Being a fairly large Working Group, its members were naturally expected to explore a variety of languages and models. The perceived benefit of the WG organization was a much broader collaboration than would otherwise be possible, in order to enhance the individual efforts, by having support for organizing short visits among partners and internal workshops gathering all partners.

Case studies

Wishing in the long run to have an impact in the real world, the WG felt it would be important to assess the practical applicability of the diverse proposals of coordination languages, models and methodologies.

To accomplish this, it was decided to set up case studies provided by the industrial partners, based on real large-scale applications and covering a wide spectrum of practical issues, to be worked upon by all partners.

Dissemination of results

To increase the downstream effects of coordination research, two major dissemination activities were planned, besides the usual submission of technical papers to conferences and journals: sponsoring a regular international conference on coordination, and publishing on the World Wide Web an up-to-date survey of the field.

Meetings

The planned meetings were five, all meant to be attended by all partners:

- An initial planning workshop, soon after the starting date.
- A kick-off meeting for the case studies, to be held within the first six months.
- Three annual workshops, to be held in the second half of each year.

The group

The group originally consisted of 13 participants: 11 academic institutions, one industrial research center (Xerox), and one industrial company (Signaal).

After the first year the WG was enlarged with the inclusion of a new partner, UCY.

The coordinator was DI–FCT/UNL.

Here is the complete list of Coordina members.

Short Name	*Institution*	*Location*	*Country*
CTH	Chalmers University of Technology	Gothenburg	Sweden
CUI	Centre Universitaire d'Informatique Université de Genève	Genève	Switzerland
CWI	Stichting Mathematisch Centrum	Amsterdam	The Netherlands
DI–FCT/UNL	Departamento de Informática Faculdade de Ciências e Tecnologia Universidade Nova de Lisboa	Monte de Caparica	Portugal
Imperial College	Imperial College of Science, Technology and Medicine	London	United Kingdom
Inria–Rennes	Institut National de Recherche en Informatique et en Automatique	Rennes	France
RUL	Rijksuniversiteit Leiden	Leiden	The Netherlands
Signaal	Hollandse Signaalapparaten	Hengelo	The Netherlands
TU Berlin	Technische Universität Berlin	Berlin	Germany
U. Bern	Universität Bern	Bern	Switzerland
UCY	University of Cyprus	Nicosia	Cyprus
UniBo	University of Bologna	Bologna	Italy
U. Pisa	University of Pisa	Pisa	Italy
Xerox	Rank Xerox Research Centre	Meylan	France

Work performed

Foundational work

We summarize the research work carried out at each of the sites, highlighting their main topics.

CTH

Gamma and Broadcasting Systems Martin Weichert worked on extending and refining the "pipelining" transformation technique that turns a producer-consumer-type Gamma program with sequential composition into a parallel one respecting the correct data flow [Wei99a]. Karol Ostrovsky, K. V. S. Prasad and Walid Taha (Yale) have developed a higher-order variant of a process calculus with broadcasting as the fundamental communication primitive [OPT00]. The concurrent message reception of broadcasting communication fits very nicely with the implicit parallelism of Gamma, a fruitful perspective for implementing a large class of Gamma programs.

A Semantic Theory for an Embedded Coordination Language David Sands, Andrew Moran and Magnus Carlsson worked on the development of a semantic theory for the non-deterministic stream processors that are at the heart of the Fudget concept. Fudgets are a collection of combinators for developing graphical user interfaces in the lazy functional programming language Haskell, which exhibit the clean separation between *computation* (the standard Haskell code) and *coordination* (the management of the user interface) that is the hallmark of coordination languages.

Security and Information Flow Andrei Sabelfeld and David Sands worked on the semantics of information flow and confidentiality [SS99], having tackled the problem of specifying and analysing secure information flow in a concurrent language for which scheduling is modelled explicitly, and is potentially probabilistic [SS00]. Johan Agat developed a combined transformation and typing system to ensure that programs do not leak information about secret data via their timing behaviour [Aga00a]. A proof-of-concept implementation for a small subset of Java bytecode is reported in [Aga00b].

CUI

The main orientation of the work at CUI was in mobile agent based Internet applications, with a strong security theme, having contributed a chapter on security [BC00] to the book "Coordination of Internet Agents".

Seal As a foundation for their agent research, the group developed the Seal calculus, an extension of the π-calculus with the notion of a bounded object or seal, representing a mobile computation or agent, with a hirearchical structure to handle security and coordination issues. The foundational work on the Seal calculus led to a PhD thesis [Vit99]. They have also developed and implemented (over JDK1.2) JavaSeal [VBB98], a mobile agent kernel that uses the Seal concepts as its basic abstractions. This was used to implement a medium sized electronic commerce application called HyperNews, that supports the super-distribution of newspaper articles using agent technology, and which ensures that payment is made whenever a user reads an article for the first time [BOV99]. The experience in implementing HyperNews led to the design of a Secure Object Space model for implementation in JavaSeal, which extends the shared space model with primitives for object locking, enabling the modeling of encryption semantics.

SecOS This is an extension of the Linda coordination model that deals with security. The SecOS model [BOV99, VBO01] includes access control primitives; data in the shared space of the system are locked with keys, and processes must possess the correct key to retrieve a data item. An implementation of SecOS was made with mechanisms for persistence and distribution, so that safe backups of the state are automatically taken and processes can access data from remote sites [Paw00]. Another direction taken was to include the model in the design of a programming language and its virtual machine for embedded and wireless network devices, to facilitate the development of applications in wireless environments, where disconnection can occur at any time [RB00]. The SecOS coordination model is particularly suitable since it enables communication without all partners having to be present on the network when the communication occurs, and it includes the necessary security mechanisms.

CWI

Manifold This is CWI's own coordination language, already implemented at the start of Coordina. It was used in the restructuring and parallelization of a number of applications [ABBE98], notably the Euler solver algorithm, the shallow water modeling and simulation application of the Delft Hydraulics Labs, and a theorem prover developed and used by another group within CWI. Preliminary work on using Manifold for distributed constraint solving led to a new four-year project funded by NWO. Practical use led to the identification and implementation of a number of language extensions. On the theoretical side, in cooperation with RUL and UniBo researchers, formal semantics of Manifold were developed, culminating in a paper [AdBB^{+}00] published in the journal of Theoretical Computer Science.

Other languages and models Along with RUL and UniBo researchers the CWI team analyzed several communication mechanisms of existing coordination models for distributed processes [BKZ99a, BKZ99b]. Both control-driven and data-driven models were considered, and it was shown that some communication mechanisms are incomparable even under a very weak notion of observation, whereas some others are equivalent with respect to a stronger notion of observation. The team looked also at some design issues for object-oriented based distributed systems, defining (with RUL and Åbo University) a notion of refinement for object-oriented systems [BKS99]. In collaboration with Utrecht University they developed a core coordination language for distributed and mobile networks of dynamically created processes [ABdB00a].

DI–FCT/UNL

TAO This is a coordination language developed in the group, based on principles of high-level abstraction and structural richness, combining control and data driven coordination features. The language-level work focussed on how to structure spaces of agents and use contextual information, and on general iterators for linking declarative search with actions. On the theory side, a powerful basic logic of actions was developed [MP98] and a range of compositional semantics was studied for a core sublanguage.

Metric-based semantics for concurrency A novel approach was developed [Mon98] that greatly simplifies the formal treatment compared with the traditional method. It led to the proposal of general notions of observation structure and observation system, and the study of the resulting categories [Mon00]. A proof was made that every nontrivial functor on the category of sets gives rise in a canonical way to a functor on the category of observation systems having a unique fixed point, an important result in giving semantics to concurrent systems.

Executable and verifiable specifications This research activity aimed at developing logical formalisms able to talk about both the temporal behavior and the spacial structure of concurrent systems. The work led to a PhD thesis [Cai99], that introduced a core programming language Lpi and the dol specification language for mobile processes. Lpi unifies, in a uniform and tightly coupled way, a deduction-oriented computation model with a concurrent and mobile model of interaction. A bridge between the logic and the Lpi model was exemplified by a dol idiom which constitutes an object-oriented specification logic that can be compiled into Lpi in a way such that encodings are models of the specifications.

Other activities A core calculus was proposed for Component-Based Programming (COP). It captures dynamic (late) binding, separates interconnection structure from components, and has an extended type system supporting abstract types and capability types for dynamic reconfiguration. A distributed programming model GroupLog was developed, to study the expressiveness, for coordination purposes, of group abstractions in modeling applications with distributed agents. In connection with the Esprit WG EuroTools, work was done in the area of Distributed Problem Solving, more specifically in Distributed Laboratories, to identify the major coordination problems and test the applicability of existing coordination models.

Imperial College

Program logics for Gamma The approach used Stone Duality to generate a program logic from a denotational semantics. The shortcoming of previous transition trace semantics was the lack of a suitable domain of multisets, a problem which was solved in the PhD thesis of Quiroz [Qui99] where a fully abstract semantics for Gamma was developed along with an associated program logic. The final part of the thesis demonstrates how the logic may be used to justify program transformations such as those suggested by

Hankin, Le Métayer and Sands [HLeMS98] and extended by Weichert [Wei99b]. This completed a theme of work started by Hankin and Gay in the Coordination project.

Static Analysis The emphasis was on semantics-based approaches, notably a pioneering one based on game semantics [MH98], which led to a good understanding of object-orientation and preliminary work on multi-threading. A general framework and an application in security were reported in [MH99]. A major manuscript on the topic of Program Analysis was published in [NRH99].

Constraint Programming The closer collaboration with Wiklicky signalled a shift in emphasis away from multi-set based views of coordination towards constraint programming. Wiklicky continued an existing collaboration with colleagues in U. Pisa, reflected in the publication list. A main result [DHW00] introduces a new notion of declarative confinement and considers security issues in the context of probabilistic concurrent constraint programming.

Inria–Rennes

The work was focussed on the study of the notions of views and coherence of views in software architecture. Coordination is seen as one relevant software architecture view, concerned with communication and synchronization issues, but many others exist, concerned with e.g. security, timeliness and fault-tolerance. The study of each view may require a different kind of decomposition of software.

The approach taken was to represent views as labelled graphs and architectural styles as graph grammars [LeM98]. Restrictions on graphs are defined by structural constraints. Inter-view consistency is defined by a correspondence relation between nodes and edges of two views, along with structural constraints similar to the internal consistency ones. Within this framework a precise language was put forward to express constraints, and a complete decision procedure was designed for this language [FLeMP99].

The team experimented with this framework by applying it to the case study proposed by Signaal [HPS97], where the views considered relevant were the functional, distribution and physical views. An example of consistency constraint is that the distribution view preserves the data flow of the functional view (a path property between the related nodes).

RUL

Action Systems The goal in this line of research was to apply this type of calculus to the problem of reasoning about coordination, i.e. find correctness-preserving refinement rules for coordination programs. First steps were taken by adding coordination primitives based on tuple spaces to Action Systems and UNITY. Then the action system framework was extended with the addition of object orientation and dynamic guards—the so-called OO-Action Systems—where one object can restrict other objects by imposing guards on them. This was an interesting new way of introducing coordination into the refinement framework, that provides a methodology for developing object-based distributed systems [BKS99, HKS00, KS00].

Foundations of Coordination Comparative studies were carried out within the field of coordination frameworks with shared distributed replicated data [BKZ99a]. Equivalence results were obtained for a number of differente software architectures. The team showed how a formal notion of components can be used in an object-oriented framework by exploiting the notion of package [tHBGK00]. They also worked on logics of domains that include infinite disjunction and conjunction [BK99].

Signaal

The research focused on the formalization of their proprietary software architecture Splice, with a view towards its potencial future replacement. Splice is an industrially developed and applied software architecture for large-scale distributed embedded systems, based on asynchronous communication between processes through a distributed data space. The characteristics of systems built with Splice include dynamic data distribution, fault-tolerance and real-time performance.

A formal model was developed to reason about the behaviour of Splice programs and to support the design process in system development. It uses a process algebra called Splice Process Algebra (SPA) [DGdJU99] that allows the derivation of properties of the architecture, from which a set of guidelines is produced to aid in the development of applications. In collaboration with RUL a transition system semantics for Splice was also developed [BKBdJ98].

The evolution of Splice was in the direction of having a logically shared data space, so that data replication does not change the semantics of the overall system [dJ99a]. Such a software architecture supporting transparent replication was in fact constructed, and operational and denotational semantics were defined for it. Among the distinguishing features of the model are idempotency of parallel composition and a delayed, inclusive choice operator on processes. The denotational semantics were defined by extending the model of state transition traces, and proofs were made of transparency of process replication and crash failure of replicas.

Signaal further worked on developing prototype middleware to support system development in accordance with their model. Since replication comes for free, fault-tolerant services can be implemented without the need for consistency protocols, a particularly important feature in time-critical applications.

TU Berlin

Coordination and the Web A substantial part of the resarch effort was in the direction of bringing together coordination and Internet technologies, with a view towards distributed coordinated applications over the Web [CTV$^+$98, Tol98b, Tol98c]. A particular area of application was that of workflow management. The adoption of coordination technologies in the Linda style led to the development of an XML/XSL based workflow system called WorkSpaces [Tol00a, Tol00b], which uses XMLSpaces to coordinate distributed workflows. XMLSpaces is an extension of Linda with multiple matching relations and distribution, that is used to store and retrieve documents transformed in steps in a coordinated manner.

Coordination at large After some work on integrating multiple coordination languages [Tol98c], the study of models of coordination [Tol00c] took a much wider view on the coordination phenomenum and how it is seen in disciplines like computer science, artificial intelligence, organizational sciences, etc.

U. Bern

The group developed a variety of tools and methodologies for supporting compositionality in component based software development, specifically targetting the design of coordination components, using cues from the domains of software architecture, scripting and object-orientation.

Piccola: a Small Composition Language Based on the π-calculus, Piccola unifies concepts found in architectural description, scripting, coordination and glue languages [ALSN00]. This line of work resulted in a PhD thesis [Lum99]. The language promotes a view of applications as scripting over components [AN00a], and was used to show how to script coordination styles [AKN00].

FLO/C: Coordination of Active Objects by Means of Explicit Connectors This is a coordination language relying on the notion of connectors, these being explicit (rule defined) entities that enforce the communication and synchronization of those entities they coordinate [DR97, Gün98].

AProcCoM: A Programmable Coordination Medium This is an open and extensible coordination medium composed of agents delivering coordination services to clients. It can be configured and dynamically extended to adapt to specific coordination requirements [Küh98].

CoLaS: Group Based Coordination for Active Objects CoLaS is a coordination language which relies on the notion of Coordination Groups. It is based on the specification and enforcement of cooperation protocols, multi-action synchronizations, and proactive behavior within groups of collaborating active objects [CD99]. Its main characteristic is the support to dynamic coordination changes.

OpenSpaces This is an object-oriented framework for reconfigurable coordination spaces [DHN00]. It supports static reconfigurability through subclassing across several dimensions, and dynamic configurability of policies through run-time composition.

UCY

The team focused on exploring the applicability of control-driven coordination, in two different areas.

E-commerce The work consisted in devising a coordination framework for electronic commerce [PA98]. The control-oriented coordination formalism was used to model e-commerce activities using event-driven coordination, and a generic framework was designed to model typical B-to-B and B-to-C transactions. The coordination paradigm allows the development of such frameworks irrespective of the details related to issues such as the payment methods advocated or the authentication protocols used.

Real-time systems The potential of using control-driven coordination for real-time systems was explored in the areas of control systems [LP00a] and distributed multimedia environments [LP00b]. The technique relied on the introduction of the notion of time in event-driven state transitions, which allows a form of temporal behaviour to be introduced into a coordinated environment.

UniBo

Theoretical analysis A significant activity took place in performing theoretical analysis of existing coordination languages in need of semantic grounding or clarification. The major focus was on Linda [BGZ97a, BGZ97b, BGZ00a, BGZ00b], with important results on differences in expressiveness linked to choices in the implementation of the output operation. Also covered were Manifold (in collaboration) [AdBB$^+$00] and JavaSpaces [BGZ00c, BGZ00d]. This gave rise to a particular study on the introduction of event notification in data-driven coordination languages [BZ00b, BZ00a].

Software Engineering Software engineering on the Internet requires revisiting traditional approaches, and the team has investigated how to tackle this issue in the light of coordination principles [BCV00, BCGZ00]. A particular attention was given to the coordination of mobile components [BCGZ00, CGR00]. The team has also proposed a new approach for the formalization of software architectures [MPL00a], which is particularly useful for automated verification.

U. Pisa

The group's many and diverse research activities can be grouped in four major topics.

Mathematical Foundations of Coordination Models and Languages The approach was based on the tile model [BMM98c], which provides a logic of concurrency with conditional state changes and synchronization. A tile-based coordination model was introduced for a calculus integrating the π-calculus with actors [FM97a, FM00, MT98]. Then came the development of zero-safe (Petri) nets [BM00b], providing algebraic mechanisms to handle coordination of transactions in a distributed setting. Later the tile logic model was extended to deal with open distributed systems, where some of the components are partially specified [BMS00].

Software Architectures A method was developed for specifying reconfigurations of software architectures, guaranteeing that if the transformation can be specified, then its application over the system will be consistent with respect to the architecture style [HIM00]. Styles are described by context-free hyperedge graph grammars.

Refinemente Calculi for Coordination The team worked on the development of a refinement framework to support the separation of functionalities from coordination policies at the right abstraction levels during the software development process. The proposed framework is called Mob_{adtl}, based on extending a Unity-like temporal logic with support for mobility of agents under the supervision of coordinating guardians [FMSS00].

Coordination Languages for Mobile Computing The language KLAIM (Kernel Language for Agents Interaction and Mobility) was introduced [BDNFP98] and both its implementation and theoretical foundations were developed. It is an experimental programming language with direct support for expressing and enforcing security policies for controlling the access to resources and data [NFP00].

Xerox

A substantial part of the research was focused on the workflow case study. The specific aspects of coordination that were studied were the challenging ones of server distribution, dynamic change and flexible task dependencies. Small demonstrators were developed for each of these features, in the framework of Xerox's WebFlow project.

Other activities included various applications of Xerox's Coordination Language Facility (CLF), notably in the domains of light-weight workflow (X-Folfer), electronic commerce (Xpect) [AP99] and document awareness (X-aware).

Case studies

The programme of the WG called for the setting up of case studies, with the purpose of bringing into sharper focus the adequacies (and problems) of particular coordination tools and methodologies in coping with practical issues of real-life applications.

The effort of devising the case studies began even before the official starting date of the project, with a major contribution from the industrial partner Signaal. In fact, for the Second International Conference on Coordination Languages and Models there was a specific call for contributions to a case study challenge prepared by Edwin de Jong from Signaal. The paper describing the challenge [dJ97] and two papers addressing it (one [HPS97] from Coordina) became part of the regular program of the conference and were published in the proceedings.

Five months after the start of Coordina a plenary meeting was held with the sole purpose of kicking off the work on the case studies. As reported elsewhere 25 people attended, representing all partners but one, and after presentations and discussions two major cases were chosen, each offering opportunities for different types of studies.

The case previously prepared by Signaal was definitely adopted, after detailed presentations that included a demo of a simplified solution using Signaal's proprietary Splice software architecture. This case study centers around the problem of real-time control of a railway network. The main challenge is that this exemplifies a class of applications where many non-functional requirements have to be addressed besides the system functionality: timeliness, fault-tolerance, degraded modes, extensibility and distribution. All of these dimensions call for the coordination philosophy of decoupling local from coordinated behaviour.

The other adopted case was based on a presentation by the other industrial partner, Xerox, around the challenges posed by a particular workflow application they were working on (bug report and repair flow in printer design and production), where then current standard worflow tools were not up to the complexities of the task. Although the workflow type of application was felt to be in good contrast and complementary to Signaal's control system case, the domain of application chosen by Xerox was felt too foreign to the majority of partners, and the area of conference management was chosen instead.

Other possible cases were discussed at the kick-off meeting, such as coordination problems in telecommunications, but the two adopted cases were considered broad and deep enough.

At the first annual workshop, five months after the case study kick-off meeting, four presentations were made on advances in the case studies, two for each case. Xerox's presentation made a convincing point that several dimensions of their original case did not find a natural mirror in the conference management case, and were in fact the most challenging aspects, not solved by standard workflow tools: server distribution, dynamic change, and flexible task dependencies.

At the third edition of the Coordination conference, even though no specific call for case study reports was made this time, two of the accepted papers were about the workflow case study [Scu99, RV99] and were submitted by Coordina members.

At the second Coordina annual workshop there were still two presentations connected to the case studies, one for each case.

Dissemination

Dissemination of the WG's efforts took many forms, which are summarized next.

The Coordination conference

One of the tasks that the WG took upon itself was to turn the International Conference on Coordination Languages and Models into a regular event. There had been a single first edition before the proposal of Coordina was made.

This task was successfully accomplished. Three editions were held under the auspices of Coordina and a steering committee left in place to keep the conference going. The interval between editions was kept at roughly one year and a half.

The second edition of the conference was held in Berlin in September 1997, and Coordina was influential in its organization although in an informal way, because its starting date was 15 August 1997. The next two events, officialy sponsored by Coordina, were held in Amsterdam in April 1999 and in Limassol (Cyprus) in September 1999.

At its first annual workshop the WG decided on a steering committee for the conference consisting solely of Coordina members: those who had been previous (co-)chairs of the conference program or organization, plus the WG coordinator. At its third annual workshop Coordina decided to change the composition of the steering committee, given that the WG was about to end, to include previous American program chairs. The program committees for each edition were always chosen, naturally, from a much broader base of researchers around the world, with two program co-chairs, one from Coordina and one from the USA. The organizing chairs were always Coordina members.

All the conference proceedings [GLeM97, CW99, PR00] were published by Springer in their series of Lecture Notes in Computer Science, widely available in most academic libraries.

The conference was an excellent vehicle to disseminate worldwide both the coordination topic and the leading role in it of European researchers, notably those in Coordina. In the 3rd and 4th editions of the conference the number of papers authored by Coordina members was, respectively, 8 in 30 and 9 in 27.

Web survey

One of the planned dissemination activities, the setting up of an up-to-date survey of Coordination on the World Wide Web, was abandoned at the first annual workshop. The main reason was the realization that the amount of effort needed for this task had been grossly underestimated, both to design and implement the site and to maintain up-to-date information. On the other hand a number of surveys had meanwhile been made available in print, so the need was no longer a strong one. Overall it was felt that putting all the group's energies into this would be a mistake, without much chance of success.

Publication of regular papers

A standard dissemination activity is of course the publication of technical papers in journals and conferences. On this account the WG performed quite well, with more than 170 publications during a period of 39 months. They are referenced at the end of this report.

Attendance of conferences and workshops

Another regular activity that helps to disseminate ideas is the attendance of conferences and workshops, not just because of the presentations to the public but also on account of the many personal exchanges that naturally take place in those occasions.

Coordina members were regularly present in many such events, helping to spread the Coordination standpoint to other communities focused on different but related topics, such as concurrency, object orientation, formal methods, static analysis, software engineering, agents, etc.

Specialized publications

A few WG members were quite active, on their own initiative, in promoting Coordination through special publications dedicated to the theme:

- a special issue [GH98] of the journal *Theoretical Computer Science*;
- a special issue [ACH98b] of the journal *Parallel Computing*, with an introductory article [ACH98a];
- another special issue [LeM99] of the journal *Theoretical Computer Science*;
- an entry on coordination languages [Han98] for the new edition of Thompson's *Encyclopedia of Computer Science and Technology.*

Specialized workshops

Coordina members also took leading roles in organizing internationl workshops on Coordination and related themes:

- 1st Workshop on Web-based Infrastrutures and Coordination Architectures for Collaborative Enterprises, at the IEEE WETICE 1999 conference, June 16/18, Stanford, USA. Robert Tolksdorf (TU Berlin) and Paolo Ciancarini (UniBo) were 2 of the 4 organizers.
- 2nd Workshop on Web-based Infrastrutures and Coordination Architectures for Collaborative Enterprises, at the IEEE WETICE 2000 conference. Robert Tolksdorf was a co-organizer.
- 1st International Workshop on Engineering Societies, at the Agent World 2000 conference. Robert Tolksdorf was a co-organizer.

External visitors

It should be mentioned that Coordina supported short stays, at various of the WG sites, of external researchers. This constitutes another form of dissemination, because such stays provide an in-depth appreciation of ongoing work, that is typically reported back at the external visitor's local environment and can influence research being done there.

Meetings

All the plenary meetings took place as planned.

Besides fulfilling the meeting programme, the WG managed to organize and hold several other significant meetings:

- two workshops on specialized topics: security and semantics;
- two bilateral colloquia between the two largest sites;
- an extra final plenary meeting, to plan for the future.

Many smaller exchanges took place, usually short visits of someone from one of the partner sites to another. All the Coordina sites were involved in this kind of activity, typically as both visitor and visited.

We next summarize the major meetings.

Initial planning workshop, Berlin (D), 1997-09-04

It took place at the Technical University of Berlin, in co-location and right after the Second International Conference on Coordination Languages and Models. All partners were represented, in a total of 19 participants.

The main decisions were 1) the fixing of dates and locations for the next two meetings, and 2) the taking up by Coordina of the reponsability over the International Conference on Coordination Languages and Models, setting up a steering committee with the previous chairs belonging to Coordina plus the WG coordinator.

Case studies kick-off meeting, Hengelo (NL), 1998-01-22/23

The meeting was held at the premises of Signaal. A total of 25 people attended, representing all partners but one. There were presentations, mainly by the industrial partners, followed by discussions.

The main decisions were 1) to work on two cases, the railroad control problem (proposed by Signaal) and the conference workflow problem (adapted from a proposal by Xerox), 2) to have particular studies within each case, and 3) to have each partner working on some of the studies.

First annual workshop, Horta (P), 1998-06-21/23

The meeting was held in a hotel. All partners were represented, totalling 26 attendees. There were 16 presentations, of which 4 concerned the case studies, and general discussion and planning at the end.

The main decisions were 1) to reinstate the original workflow problem proposed by Xerox as a proper case study, 2) to abandon the goal of building a Coordination survey on the Web, and 3) to co-locate the second annual workshop with the next Coordination conference.

First "Pisa meets Amsterdam" colloquium, Pisa (I), 1999-01-21/22

It was held at the University of Pisa, with 12 participants, mostly from the two partners U. Pisa and CWI. There were 9 presentations, followed by discussion of collaboration plans.

Second annual workshop, Amsterdam (NL), 1999-04-28/29

The meeting was held at the premises of CWI, in co-location and right after the Third International Conference on Coordination Languages and Models. There were 35 participants representing all partners. The program consisted of 8 presentations and a final planning session.

The main decisions were 1) the tentative dates and location of the next Coordination conference, 2) to co-locate the third annual workshop with the conference (implying a needed WG extension), and 3) to hold two specialized workshops, one on security at CTH and the other on semantics at Imperial College.

Workshop on security, Göteborg (S), 1999-06-10/11

Held at Chalmers (CTH), it brought together a total of 16 participants, mostly from both Coordina (3 sites) and a Swedish NUTEK PROMODIS project cluster. A total of 10 presentations were made, half of them from Coordina participants.

Second "Pisa meets Amsterdam" colloquium, Amsterdam (NL), 2000-01-13/14

Held at CWI this time, the meeting brought together 13 participants from the two sites U. Pisa and CWI. There were 9 presentations.

Workshop on semantics, London (UK), 2000-04-13/14

The meeting took place at Imperial College, and was attended by 15 people mostly from Coordina (5 sites). The program consisted of 7 technical presentations.

Third annual workshop, Limassol (CY), 2000-09-14

It was held in the same hotel and right after the Fourth International Conference on Coordination Languages and Models. It was attended by 12 people from 11 Coordina sites. There were just 4 technical presentations, after discussions about the future of both the conference and the group.

The main decisions were 1) the time and place of the next Coordination conference, 2) the terms of renewal of its steering committee, 3) to hold a final workshop to discuss in detail new collaborative actions, and 4) to invite to that workshop a few active researchers outside Coordina.

Final workshop, Amsterdam (NL), 2000-11-6/7

The meeting was held at CWI. There were 21 participants, of which 16 were from 10 Coordina sites and 5 were invited outsiders, representing 4 academic sites—Univ. of Bologna (Engineering), Univ. of Lisbon, Univ. of Utrecht and Univ. of York—and 2 companies— Oblog (P) and Ordina (NL).

The meeting started with an initial roundup of research interests and plans, continued with a discusssion of the available actions supported by the EC, split into submeetings on four identified themes of common interest, and concluded with a final methodological discussion.

The main decisions were 1) to concentrate on a joint proposal for a research and training network, 2) to have UniBo as the coordinator site, and 3) to name the new proposal CooRTEx—Coordination Research and Training Exchange.

Results

Overall the WG activities can be seen as quite successful.

The major achievement is the firm consolidation, worldwide, of the term “Coordination”, what it stands for in Computer Science—the methodological standpoint of separating non-local coordination phenomena from local functionalities—and the richness of technical content of the field. Along with this consolidation has come the recognition of the leading role of European research in this area, notably that originating from the Coordina sites.

Other noticeable Coordina results are a growing web of connections of Coordination to other fields, concrete evidence of its applicability, and a much better link between its theory and practice.

Consolidation

Before the start of Coordina the field of Coordination was not yet mature. There had been the pioneering Esprit project 9102 “Coordination” (from which came a sizable part of Coordina) and scattered efforts in the non-European world. The end of that project coincided with the First International Conference on Coordination Languages and Models, held at Cesena (Italy) in April 1996, where for the first time Coordination as a theme found a global forum.

The Coordina WG produced a very active collaborative momentum, that was very influential in bringing the field into maturity, through the combination of widespread research, continuously cross-checked by internal workshops and mutual visits, with a concerted dissemination effort of publishing and turning the Coordination conference into a successful regular event.

At a time when many conferences saw their importance dwindling and had to group together in larger events, the vitality of Coordination was evident in the ability to hold three editions of the conference with a good submission rate, high quality of selected papers, and sizable if not large attendance. These conference editions were certainly instrumental in the worldwide recognition of Coordination as a proper field in Computer Science.

Richness

The richness of the technical content of the field can be glimpsed from the preceding summaries of research activity at each Coordina site. One of the results of this research was the ability to overcome an initial prejudice that narrowly associated Coordination with just a particular class of models based on tuple space interaction, such as Linda or Gamma, that the WG came to view as endogenous[1] data-driven[2] models.

The range of basic models was extended to exogenous[3] and control-driven[4], and many concrete language proposals explored these possibilities, alone or in combinations. Not only did these basic models gain full status, other more specialized models emerged trying to integrate coordination with other concerns such as mobility and security. Another important issue that was also explored in the search for linguistic support was that of compositionality.

Connections

Rather than maintaining the coordination theme closed on itself, the WG managed to establish many connections to other fields and themes in Computer Science, in fact advancing the awareness in those other communities of the broad, orthogonal role Coordination is bound to play. Many examples can be found in this report, both in the description of research activities and in the references, of established connections to object and component oriented programming, concurrency theory, software engineering, fault-tolerance, distributed and mobile computing, Internet technology, etc.

[1] Local functionalities invoke coordination actions.

[2] Coordination works by setting and testing of data.

[3] Coordination procedures handle local functionalities.

[4] Coordination actions directly affect the process structure.

Applicability

As expected, the case studies allowed a more focused appreciation of a number of characteristics of several coordination approaches. The main split between the basic data-driven and control-driven models proved to be linked to different types of applications, the former being more natural for the kind of loose-coupled activity exhibited in the control system case whereas the latter fits more naturally into the tighter-coupled workflow activity of the other case. But, on the other hand, it was shown that such mappings are in reality never strict, as there were aspects of either case where the reverse suitability of models was found.

In the output of publications from the WG there is ample evidence of the real applicability of coordination, coming not only from the adopted case studies, as already mentioned, but also from other application areas, such as parallel algorithms (e.g. for fluid dynamics), electronic commerce or financial applications on the Web.

Theory and Practice

An important outcome of the WG was a good coupling of theoretical research with practical concerns.

Several theoretical studies were carried out to clarify semantic properties of existing or proposed languages or systems, many originating from outside Coordina, and interesting results have shown practical implications of certain implementation choices that were taken or sometimes left unspecified. For example, there was a characterization of differences in expressive power achieved by different timing characteristics of the output operation in Linda, and some corresponding features in JavaSpaces. As another example, the preocupation with a theoretically sound model led to an important practical improvement of Splice to achieve transparent replication.

Results of theoretical studies with practical relevance were also produced in two other different areas: static analysis and verification of important properties (e.g. security) and refinement procedures for stepwise software development at adequate abstraction levels.

Conclusions

Looking back at the work and achievements of the Coordina WG, the main conclusions are the following.

- Coordination was put firmly in the map of Computer Science.
- The applicability of coordination was convincingly demonstrated.
- A rich range of coordination models was exposed in the scientific litterature.
- No basic coordination model wins over the others. All find their natural suitability to different aspects of many applications, and are thus complementary to one another.
- Although the case studies have shown partial solutions, they have also made it clear that there are still unanswered difficulties, specially in coherently tackling the many non-functional views of real systems.
- Coordination was shown to have an impact on many and varied other fields, rather than being a confined topic.
- In coordination, as in other fields, theoretical studies have shown their practical usefulness, in highlighting and making clear the implications of certain implementation choices, and in providing techniques for helping in software development.
- Although coordination research has reached worldwide status, Europe is still very much the leader in the field. The number and spread of European researchers on coordination has grown, thanks in part to the existence of Coordina.
- The level of maturity and dissemination of the field promoted by Coordina suggests the opportunity of turning into mainstream practice the use of coordination tools and techniques.
- To help achieve the previous goal, some of the former Coordina members, along with some of the more active new players in the field, have decided to submit a proposal for a European research and training network.

References

[ABBE98] F. Arbab, C. L. Blom, F. J. Burger, and C. T. H. Everaars. Reusable coordinator modules for massively concurrent applications. *Software: Practice and Experience*, 28(7):703–735, June 1998.

[ABdB00a] F. Arbab, M.M. Bonsangue, and F.S. de Boer. A coordination language for mobile components. In J. Carrol, E. Damiani, H. Haddad, and D. Oppenheim, editors, *Proceedings of the 2000 ACM Symposium on Applied Computing (SAC 2000)*, pages 166–173. ACM Press, 2000.

[ABdB00b] F. Arbab, M.M. Bonsangue, and F.S. de Boer. A logical interface description language for components. In A. Porto and G.-C. Roman, editors, *Coordination Languages and Models, 4th International Conference*, volume 1906 of *Lecture Notes in Computer Science*, pages 249–266. Springer, 2000.

[ACH98a] F. Arbab, P. Ciancarini, and C. Hankin. Coordination languages for parallel programming. *Parallel Computing*, 24:989–1004, 1998.

[ACH98b] F. Arbab, P. Ciancarini, and C. Hankin. Editors, special issue on coordination languages. *Parallel Computing*, 24, 1998.

[Ach00] Franz Achermann. Language support for feature mixing. In *Workshop on Multi-Dimensional Separation of Concerns in Software Engineering (ICSE 2000)*, Limerick, Ireland, June 2000.

[AdBB$^+$00] Farhad Arbab, Jaco W. de Bakker, Marcello M. Bonsangue, Jan J.M.M. Rutten, Adriano Scutellá, and Gianluigi Zavattaro. A transition system semantics for the control-driven coordination language manifold. *Theoretical Computer Science*, 240(1):3–47, June 2000.

[Aga00a] J. Agat. Transforming out timing leaks. In *POPL'00: The 27:th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 40–53. ACM Press, January 2000.

[Aga00b] Johan Agat. Transforming out timing leaks in practice. Technical report, Chalmers University of Technology, 2000.
`http://www.cs.chalmers.se/agat/Publications/paperII.ps.gz`.

[AKN00] Franz Achermann, Stefan Kneubuehl, and Oscar Nierstrasz. Scripting coordination styles. In António Porto and Gruia-Catalin Roman, editors, *Coordination Languages and Models, 4th International Conference*, volume 1906 of *Lecture Notes in Computer Science*, pages 19–35. Springer, September 2000.

[ALSN00] Franz Achermann, Markus Lumpe, Jean-Guy Schneider, and Oscar Nierstrasz. Piccola – a small composition language. In Howard Bowman and John Derrick., editors, *Formal Methods for Distributed Processing, an Object Oriented Approach.* Cambridge University Press, 2000.

[AM98] F. Arbab and E. Monfroy. Coordination of heterogeneous distributed cooperative constraint solving. *Applied Computing Review*, 6(2):4–17, 1998.

[AN00a] Franz Achermann and Oscar Nierstrasz. Applications = Components + Scripts – A Tour of Piccola. In Mehmet Aksit, editor, *Software Architectures and Component Technology.* Kluwer, 2000.

[AN00b] Franz Achermann and Oscar Nierstrasz. Explicit Namespaces. In Jürg Gutknecht and Wolfgang Weck, editors, *Modular Programming Languages*, volume 1897 of *Lecture Notes in Computer Science*, pages 77–89, Zurich, Switzerland, September 2000.

[AP99] Jean-Marc Andreoli and François Pacull. Distributed print on demand systems in the xpect framework. *Journal of Distributed and Parallel Databases*, 7(2), April 1999.

[Arb99] F. Arbab. Coordination programming for parallel and distributed applications. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '99)*, volume 1, pages 1–10, Las Vegas, Nevada, 1999.

[BAS00] Pascal Bouvry, Farhad Arbab, and Franciszek Seredynski. Distributed evolutionary optimization in manifold: Rosenbrock's function case study. *Information Sciences*, 122:141–159, 2000.

[BC00] Ciaran Bryce and Marco Cremonini. Coordination and Security on the Internet. In Omicini and Zambonelli, editors, *Coordination of Internet Agents*. Springer, 2000.

[BCGZ00] N. Busi, P. Ciancarini, R. Gorrieri, and G. Zavattaro. Models for Coordinating Agents: a Guided Tour. In A. Omicini, R. Tolksdorf, G. Weiss, and F. Zambonelli, editors, *Coordination models and applications for agents*. Springer, 2000.

[BCV00] L. Bompani, P. Ciancarini, and F. Vitali. Software Engineering on the Internet: A Roadmap. In A. Finkelstein, editor, *Proc. 22nd Intl. Conf. on Software Engioneering - Track "The Future of Software Engineering"*, pages 303–318, Limerick, Ireland, 2000.

[BdFEMOM00] Roberto Bruni, David de Frutos-Escrig, Narciso Martí-Oliet, and Ugo Montanari. Bisimilarity congruences for open terms and term graphs via tile logic. In Catuscia Palamidessi, editor, *Proceedings of CONCUR 2000, 11th International Conference on Concurrency Theory*, volume 1877 of *Lecture Notes in Computer Science*, pages 259–274. Springer, 2000.

[BDNFP98] L. Bettini, R. De Nicola, G. Ferrari, and R. Pugliese. Interactive mobile agents in XKLAIM. In *Proceedings of WETICE '98*. IEEE, Computer Society Press, 1998.

[BGZ97a] N. Busi, R. Gorrieri, and G. Zavattaro. On the Turing equivalence of Linda coordination primitives. In *Proc. 4th Workshop on Expressiveness in Concurrency*, volume 7 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1997.

[BGZ97b] N. Busi, R. Gorrieri, and G. Zavattaro. Three semantics of the output operation for generative communication. In David Garlan and Daniel Le Métayer, editors, *Coordination Languages and Models, Second International Conference*, volume 1282 of *Lecture Notes in Computer Science*, pages 205–219. Springer, 1997.

[BGZ00a] Nadia Busi, Roberto Gorrieri, and Gianluigi Zavattaro. Comparing three semantics for Linda-like languages. *Theoretical Computer Science*, 240(1):49–90, June 2000.

[BGZ00b] Nadia Busi, Roberto Gorrieri, and Gianluigi Zavattaro. On the expressiveness of Linda coordination primitives. *Information and Computation*, 156(1-2):90–121, January 2000.

[BGZ00c] Nadia Busi, Roberto Gorrieri, and Gianluigi Zavattaro. On the semantics of JavaSpaces. In *Proc. Fourth IFIP Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'00)*, pages 3–19. Kluwer Academic, September 2000.

[BGZ00d] Nadia Busi, Roberto Gorrieri, and Gianluigi Zavattaro. Process calculi for coordination: from Linda to JavaSpaces. In *Proc. Eighth International Conference on Algebraic Methodology and Software Technology (AMAST'00)*, volume 1816 of *Lecture Notes in Computer Science*, pages 198–212. Springer, May 2000.

[BK97a] M.M. Bonsangue and J.N. Kok. Infinitary domain logic for finitary transition systems. In M. Abadi and T. Ito, editors, *Proceedings of the 3rd Symposium on Theoretical Aspects of Computer Science (TACS '97)*, volume 1281 of *Lecture Notes in Computer Science*, pages 213–232. Springer, 1997.

[BK97b] M.M. Bonsangue and J.N. Kok. Specifying computations using hyper transition systems. In I. Privara and P. Ruzicka, editors, *Proceedings of the 22nd Internationl Symposium on Mathematical Foundations of Computer Science (MFCS '97)*, volume 1295 of *Lecture Notes in Computer Science*, pages 169–178. Springer, 1997.

[BK99] M.M. Bonsangue and J.N. Kok. Towards an infinitary logic of domains: Abramsky logic for transition systems. *Information and Computation*, 155:170–201, 1999.

[BKBdJ98] M.M. Bonsangue, J.N. Kok, M. Boasson, and E. de Jong. A software architecture for distributed control systems and its transition system semantics. In J. Carroll, J.B. Lamont, D. Oppenheim, K.M. George, and B. Bryant, editors, *Proceedings of the 1998 ACM Symposium on Applied Computing (SAC'98)*, pages 159–168. ACM Press, 1998.

[BKS98] M.M. Bonsangue, J.N. Kok, and Kaisa Sere. An approach to object-orientation in action systems. In J. Jeuring, editor, *Proceedings of the 4th Internationl Conference on Mathematics of Program Construction (MPC '98)*, volume 1422 of *Lecture Notes in Computer Science*, pages 68–95. Springer, 1998.

[BKS99] M.M. Bonsangue, J.N. Kok, and K. Sere. Developing object-based distributed systems. In P. Ciancarini, A. Fantechi, and R. Gorrieri, editors, *Proceedings of the 3rd IFIP International Conference on Formal Methods for Open Object-based Distributed Systems (FMOODS'99)*, pages 19–34. Kluwer, 1999.

[BKZ99a] M.M. Bonsangue, J.N. Kok, and G. Zavattaro. Comparing coordination models based on shared distributed replicated data. In J. Carroll, H. Haddad, D. Oppenheim, B. Bryant, and G.B. Lamont, editors, *Proceedings of the 1999 ACM Symposium on Applied Computing (SAC '99)*, pages 146 – 155. ACM Press, 1999.

[BKZ99b] M.M. Bonsangue, J.N. Kok, and G. Zavattaro. Comparing software architectures for coordination languages. In P. Ciancarini and A. Wolf, editors, *Coordination Languages and Models (Proceedings of the 3rd international conference)*, volume 1594 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 1999.

[BKZ00] M.M. Bonsangue, J.N. Kok, and G. Zavattaro. Comparing coordination models and architectures using embeddings. Technical report SEN-R0025, CWI, Amsterdam, 2000.

[BM98] R. Bruni and U. Montanari. Zero-safe nets: The individual token approach. In F. Parisi-Presicce, editor, *Proceedings of WADT'97, 12th workshop on Recent Trends in Algebraic Development Techniques*, volume 1376 of *LNCS*, pages 122–140. Springer Verlag, 1998.

[BM00a] Roberto Bruni and Ugo Montanari. Executing transactions in zero-safe nets. In Mogens Nielsen and Dan Simpson, editors, *Proceedings of ICATPN2000, 21st International Conference on Application and Theory of Petri Nets*, volume 1825 of *Lecture Notes in Computer Science*, pages 83–102. Springer, 2000.

[BM00b] Roberto Bruni and Ugo Montanari. Zero-safe nets: Comparing the collective and individual token approaches. *Information and Computation*, 156:46–89, 2000.

[BMM98a] R. Bruni, J. Meseguer, and U. Montanari. Implementing tile systems: Some examples from process calculi. In P. Degano, U. Vaccaro, and G. Pirillo, editors, *Proceedings of ICTCS'98, 6th Italian Conference on Theoretical Computer Science*, pages 168–179. World Scientific, 1998.

[BMM98b] R. Bruni, J. Meseguer, and U. Montanari. Internal strategies in a rewriting implementation of tile systems. In C. Kirchner and H. Kirchner, editors, *Proceedings of WRLA'98, 2nd workshop on Rewriting Logic and its Applications*, volume 15 of *ENTCS*. Elsevier, 1998.

[BMM98c] R. Bruni, J. Meseguer, and U. Montanari. Process and term tile logic. Technical Report SRI-CSL-98-06, SRI International, 1998. Also Technical Report TR-98-09, Computer Science Department, University of Pisa.

[BMMS98] R. Bruni, J. Meseguer, U. Montanari, and V. Sassone. A comparison of petri net semantics under the collective token philosophy. In J. Hsiang and A. Ohori, editors, *Proceedings of ASIAN'98, 4th Asian Computing Science Conference*, volume 1538 of *LNCS*, pages 225–244. Springer Verlag, 1998.

[BMS00] Roberto Bruni, Ugo Montanari, and Vladimiro Sassone. Open ended systems, dynamic bisimulation and tile logic. In Jan van Leeuwen, Osamu Watanabe, Masami Hagiya, Peter D. Mosses, and Takayasu Ito, editors, *Proceedings of IFIP TCS2000, IFIP International Conference on Theoretical Computer Science*, volume 1872 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2000.

[BOV99] C. Bryce, M. Oriol, and J. Vitek. A coordination model for agents based on secure spaces. In P. Ciancarini and A. Wolf, editors, *Coordination Languages and Models (Proceedings of the 3rd international conference)*, volume 1594 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 1999.

[Bru98] R. Bruni. A logic for modular descriptions of asynchronous and synchronized concurrent systems. In C. Kirchner and H. Kirchner, editors, *Proceedings of WRLA'98, 2nd workshop on Rewriting Logic and its Applications*, volume 15 of *ENTCS*. Elsevier, 1998.

[BS00] Roberto Bruni and Vladimiro Sassone. Algebraic models for contextual nets. In Ugo Montanari, José D.P. Rolim, and Emo Welzl, editors, *Proceedings of ICALP2000, 27th International Colloquium on Automata, Languages and Programming*, volume 1853 of *Lecture Notes in Computer Science*, pages 175–186. Springer, 2000.

[BZ00a] Nadia Busi and Gianluigi Zavattaro. Event notification in data-driven coordination languages: Comparing the ordered and unordered interpretations. In *Proc. 2000 ACM Symposium on Applied Computing (SAC'00)*, pages 233–239. ACM Press, March 2000.

[BZ00b] Nadia Busi and Gianluigi Zavattaro. On the expressiveness of event notification in data-driven coordination languages. In *Proc. 2000 European Symposium on Programming (ESOP'00)*, volume 1782 of *Lecture Notes in Computer Science*, pages 41–55. Springer, March 2000.

[Cai99] Luís Caires. *A Model for Declarative Programming and Specification with Concurrency and Mobility*. PhD thesis, Universidade Nova de Lisboa, 1999.

[CD99] Juan Carlos Cruz and Stéphane Ducasse. A group based approach for coordinating active objects. In P. Ciancarini and A. Wolf, editors, *Coordination Languages and Models (Proceedings of the 3rd international conference)*, volume 1594 of *Lecture Notes in Computer Science*, pages 355–370. Springer, 1999.

[CFM00] P. Ciancarini, F. Franzè, and C. Mascolo. Using a Coordination Language to Specify and Analyze Systems Containing Mobile Components. *ACM Transactions on Software Engineering and Methodology*, 9(2):167–198, 2000.

[CGR00] P. Ciancarini, A. Giovannini, and D. Rossi. Mobility and Coordination for Distributed Java Applications. In S. Krakowiak and S. Shrivastava, editors, *Recent Advances in Distributed Systems*, volume 1752 of *Lecture Notes in Computer Science*, pages 402–425. Springer, 2000.

[CKR$^+$97] Paolo Ciancarini, Andreas Knoche, Davide Rossi, Robert Tolksdorf, and Fabio Vitali. Coordinating java agents for financial applications on the www. In *Proceedings of the The Second International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, PAAM'97*, pages 179–193, 1997.

[CM97] Luís Caires and Luís Monteiro. Proof net semantics of proof search computation. In K. Meinka and M. Hanus, editors, *Proceedings of the Algebraic and Logic Programming International Conference*, volume 1298 of *Lecture Notes in Computer Science*, pages 194–208. Springer, 1997.

[CM98] Luís Caires and Luís Monteiro. Verifiable and executable specifications of concurrent objects in $\mathcal{L}_\pi$. In C. Hankin, editor, *Programming Languages and Systems, Proc. ESOP'98*, volume 1381 of *Lecture Notes in Computer Science*, pages 42–56. Springer, 1998.

[CT98] Juan Carlos Cruz and Sander Tichelaar. Managing evolution of coordination aspects in open systems. In *Proceedings of the DEXA workshops*, pages 578–582. IEEE Computer Society Press, 1998.

[CTV$^+$98] Paolo Ciancarini, Robert Tolksdorf, Davide Vitali, Fabio Rossi, and Andreas Knoche. Coordinating multiagent applications on the WWW: a reference architecture. *IEEE Transactions on Software Engineering*, 24(5):362–366, 1998.

[CTV99] P. Ciancarini, R. Tolksdorf, and F. Vitali. The World Wide Web as a Place for Agents. In M. Woolridge and M. Veloso, editors, *Artificial Intelligence Today. Recent Trends and Developments*, volume 1600 of *Lecture Notes in Artificial Intelligence*, pages 175–194. Springer, 1999.

[CW99] P. Ciancarini and A. Wolf, editors. *Coordination Languages and Models, Third International Conference*, volume 1594 of *Lecture Notes in Computer Science*. Springer, 1999.

[CW00] P. Ciancarini and M. Wooldridge, editors. *First Int. ICSE Workshop on Agent Oriented Software Engineering*, volume 1957 of *Lecture Notes in Computer Science*, Limerick, Ireland, 2000. Springer.

[Dam98] L. Dami. A lambda calculus for dynamic binding. *Theoretical Computer Science*, 192(2), 1998.

[dBB00a] F.S. de Boer and M.M. Bonsangue. A compositional model for confluent dynamic dataflow networks. Technical report SEN-R0021, CWI, Amsterdam, 2000.

[dBB00b] F.S. de Boer and M.M. Bonsangue. A compositional model for dynamical data-flow networks. In M. Nielsen and B. Rovan, editors, *Proceedings of the 25nd MFCS, Bratislava, Slovakia*, volume 1893, pages 212–221. Springer, 2000.

[dBvB00] J.W. de Bakker and F.C. van Breugel. From Banach to Milner: Metric semantics for second order communication and concurrency. In G.D. Plotkin et al, editor, *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, pages 99–132. MIT Press, 2000.

[DG98] Stéphane Ducasse and Manuel Günter. Coordination of active objects by means of explicit connectors. In *Proceedings of the DEXA workshops*, pages 572–577. IEEE Computer Society Press, 1998.

[DGdJU99] P. Dechering, R. Groenboom, E. de Jong, and J.T. Udding. Formalization of a software architecture for embedded systems: a process algebra for splice. In *Proceedings of the Hawaiian International Conference on System Sciences (HICSS-32)*. IEEE Computer Society, 1999.

[DHN00] Stéphane Ducasse, Thomas Hofmann, and Oscar Nierstrasz. Openspaces: An object-oriented framework for reconfigurable coordination spaces. In António Porto and Gruia-Catalin Roman, editors, *Coordination Languages and Models, 4th International Conference*, volume 1906 of *Lecture Notes in Computer Science*, pages 1–19. Springer, September 2000.

[DHW00] Alessandra Di Pierro, Chris Hankin, and Herbert Wiklicky. Probabilistic security analysis in a declarative framework. In P. Degano, editor, *Proceedings of WITS'00 – Workshop on Issues in the Theory of Security, 7-8 July,Geneva, Switzerland*, pages 83–86, July 2000.

[dJ97] Edwin de Jong. Software architecture for large control systems. In David Garlan and Daniel Le Métayer, editors, *Coordination Languages and Models, Second International Conference*, volume 1282 of *Lecture Notes in Computer Science*, pages 150–156. Springer, 1997.

[dJ99a] E. de Jong. Distributed shared data space architecture for real-time systems. In *Proc. 2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'99)*, pages 159–162. IEEE Computer Society Press, 1999.

[dJ99b] E. de Jong. Distributed shared data space architecture for real-time systems. In *Proceedings 2nd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'99)*, pages 159–162. IEEE Computer Society, 1999.

[DNFP97] R. De Nicola, G. Ferrari, and R Pugliese. Coordinating mobile agents via blackboards and access rights. In *Proceedings of Coordination'97*, volume 1282 of *LNCS*. Springer Verlag, 1997.

[DNFP98] R. De Nicola, G. Ferrari, and R. Pugliese. Klaim: a kernel language for agents interaction and mobility. *IEEE Transactions on Software Engineering*, 24:315–330, 1998.

[DR97] Stéphane Ducasse and Tamar Richner. Executable connectors: towards reusable design elements. In *Proceedings of EXEC/FSE'97*, volume 1301 of *Lecture Notes in Computer Science*, pages 483–500. Springer, 1997.

[dVR97] E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. In *Proceedings of ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 460–470. Springer, 1997.

[dVR99] E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theoretical Computer Science*, 221(1-2):271–293, 1999.

[DW99] Alessandra Di Pierro and Herbert Wiklicky. Quantitative observables, averages and constraint programming. In K.R. Apt, T. Kakas, E. Monfroy, and F. Rossi, editors, *Proceedings of the ERCIM/Compulog Workshop on Constraints, 25-27 October, Paphos, Cyprus*, Nicosia, Cyprus, 1999. University of Cyprus.

[DW00a] Alessandra Di Pierro and Herbert Wiklicky. Formalising randomised algorithms in constraint programming. In K.R. Apt, E. Monfroy, and F. Rossi, editors, *Proceedings of the ERCIM/Compulog Workshop on Constraints, 19-21 June, Padova, Italy*, Padova, Italy, 2000. Universitá di Padova.

[DW00b] Alessandra Di Pierro and Herbert Wiklicky. Quantitative observables and averages in Probabilistic Concurrent Constraint Programming. In K.R. Apt, T. Kakas, E. Monfroy, and F. Rossi, editors, *New Trends in Constraints — Selected Papers of the ERCIM/Compulog Workshop on Constraints, October 1999, Paphos, Cyprus*, number 1865 in Lecture Notes in Computer Science, Berlin — Heidelberg — New York, 2000. Springer Verlag.

[EAK99] K. Everaars, F. Arbab, and B. Koren. Dynamic process composition and communication patterns in irregularly structured applications. In *Proceedings of Irregular'99 (San Juan, Puerto Rico)*, Lecture Notes in Computer Science. Springer, 1999.

[EAK00a] C.T.H. Everaars, F. Arbab, and B. Koren. Parallel, distributed-memory implementation of sparse-grid methods for three-dimensional fluid-flow computations. Technical report SEN-R0039, CWI, Amsterdam, 2000.

[EAK00b] K. Everaars, F. Arbab, and B. Koren. Dynamic process composition and communication patterns in irregularly structured applications. *Concurrency: Practice and Experience*, 12:157–174, 2000.

[Err99] D. L. Errington. *Twisted Systems.* PhD thesis, University of London, 1999.

[FLeM98] P. Fradet and D. Le Métayer. Structured gamma. *Science of Computer Programming*, 31(2-3):263–289, 1998.

[FLeMP99] P. Fradet, D. Le Métayer, and M. Périn. Consistency checking for multiple view software architectures. In *7th ACM SIGSOFT Symp. on the Foundations of Software Engineering*, Lecture Notes in Computer Science, pages 410,428. Springer-Verlag, September 1999.

[FM97a] G. Ferrari and U. Montanari. A tile-based coordination view of the asynchronous π-calculus. In *Proceedings of MFCS'97*, volume 1295 of *LNCS*. Springer Verlag, 1997.

[FM97b] G. Ferrari and U. Montanari. Tiles for concurrent and located calculi. In C. Palamidessi and J. Parrow, editors, *Proceedings of EXPRESS'97, 4th workshop on Expressiveness in Concurrency*, volume 7 of *ENTCS*. Elsevier, 1997.

[FM00] G. Ferrari and U. Montanari. Tiles for located and mobile processes. *Information and Computation*, 156, 2000.

[FMSS00] G. Ferrari, C. Montangero, L. Semini, and S. Semprini. Mobile agents coordination in mob_{adtl}. In A. Porto and G.-C. Roman, editors, *Coordination Languages and Models, 4th International Conference*, volume 1906 of *Lecture Notes in Computer Science*. Springer, 2000.

[GH98] Roberto Gorrieri and Chris Hankin. Editors, special issue on coordination languages. *Theoretical Computer Science*, 192, 1998.

[GKSU98] H. Goeman, J.N. Kok, K. Sere, and R.T. Udink. Coordination in the InpUnity framework. *Science of Computer Programming*, 31:313–334, 1998.

[GLeM97] David Garlan and Daniel Le Métayer, editors. *Coordination Languages and Models, Second International Conference*, volume 1282 of *Lecture Notes in Computer Science*. Springer, 1997.

[Gün98] Manuel Günter. Explicit connectors for coordination of active objects. Diploma thesis, University of Bern, March 1998.

[GZ99] M. Gaspari and G. Zavattaro. An algebra of actors. In P. Ciancarini, A. Fantechi, and R. Gorrieri, editors, *Proc. 3rd IFIP Workshop on Formal Methods for Open Object-based Distributed Systems (FMOODS)*, pages 3–18, 1999.

[Han98] Chris Hankin. Coordination languages. In *The Encyclopedia of Computer Science and Technology*. International Thompson, 1998.

[HIM00] D. Hirsh, P. Inverardi, and U. Montanari. Reconfiguration of software architecture styles with name mobility. In A. Porto and G.-C. Roman, editors, *Coordination Languages and Models, 4th International Conference*, volume 1906 of *Lecture Notes in Computer Science*. Springer, 2000.

[HKS00] E. Hedman, J.N. Kok, and K. Sere. Coordinating action systems. *Theoretical Computer Science*, 240:91–115, 2000.

[HLeMS98] C. Hankin, D. Le Métayer, and D. Sands. Refining multiset transformers. *Theoretical Computer Science*, 192, January 1998.

[HPS97] A. A. Holzbacher, M. Périn, and M. Südholt. Modeling railway control systems using graph grammars: a case study. In David Garlan and Daniel Le Métayer, editors, *Coordination Languages and Models, Second International Conference*, volume 1282 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 1997.

[JLeMT99] T. Jensen, D. Le Métayer, and T. Thorn. Verification of control flow based security properties. In *Proc. of the 20th IEEE Symp. on Security and Privacy*, pages 89–103. New York: IEEE Computer Society, May 1999.

[JR99] B. Jacobs and J.J.M.M. Rutten, editors. *Proceedings of the Second Workshop on Coalgebraic Methods in Computer Science (CMCS '99)*, volume 19 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1999.

[KK97] P. Knijnenburg and J.N. Kok. The semantics of the combination of atomized statements and parallel choice. *Formal Aspects of Computing*, 9:518–536, 1997.

[KS00] J.N. Kok and K. Sere. Coordination-based design of distributed systems. In *Proceedings Coordination Languages and Models*, number 1906 in Lecture Notes in Computer Science, pages 347–353. Springer Verlag, 2000.

[Küh98] Daniel Kühni. AProCoM: A programmable coordination medium. Diploma thesis, University of Bern, October 1998.

[LAN00] Markus Lumpe, Franz Achermann, and Oscar Nierstrasz. A Formal Language for Composition. In Gary Leavens and Murali Sitaraman, editors, *Foundations of Component Based Systems*, pages 69–90. Cambridge University Press, 2000.

[LeM98] Daniel Le Métayer. Describing software architecture styles using graph grammars. *IEEE Transactions on Software Engineering*, 24:521–533, July 1998.

[LeM99] D. Le Métayer. Editor, special issue on coordination and mobility. *Theoretical Computer Science*, 240(1), 1999.

[LP00a] T. A. Lemniotes and G. A. Papadopoulos. Modelling control systems in an event-driven coordination language. In *Proc. Eighth International Conference on High Performance Computing and Networking Europe (HPCN 2000)*, volume 1823 of *Lecture Notes in Computer Science*, pages 177–186. Springer, 2000.

[LP00b] T. A. Lemniotes and G. A. Papadopoulos. Real-time coordination in distributed multimedia systems. In *Proc. 14th International Parallel and Distributed Processing Symposium (IPDPS 2000), Eighth International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS 2000)*, volume 1800 of *Lecture Notes in Computer Science*, pages 685–691. Springer, 2000.

[LTM00] Giovani Librelotto, Simão Toscani, and Luís Monteiro. Distribuição da linguagem RS numa plataforma MDX. In *Simpósio Brasileiro de Informática, SBI'2000*, 2000. In Portuguese.

[Lum99] Markus Lumpe. *A Pi-Calculus Based Approach to Software Composition.* Ph.D. thesis, University of Bern, Institute of Computer Science and Applied Mathematics, January 1999.

[Mas99] C. Mascolo. MobiS: A Specification Language for Mobile Systems. In P. Ciancarini and A. Wolf, editors, *Proc. 3rd Int. Conf. on Coordination Models and Languages*, volume 1594 of *Lecture Notes in Computer Science*, pages 37–52, Amsterdam, Netherland, April 1999. Springer.

[MH98] P. Malacaria and C. Hankin. Generalised flowcharts and games. In *Proceedings of ICALP'98*, volume 1443. Springer, 1998.

[MH99] P. Malacaria and C. Hankin. Nondeterministic games and program analysis: An application to security. In *Proceedings of LICS'99.* IEEE Press, 1999.

[MM98] U. Montanari and J. Meseguer. Mapping tile logic into rewriting logic. In F. Parisi-Presicce, editor, *Proceedings of WADT'97, 12th workshop on Recent Trends in Algebraic Development Techniques*, volume 1376 of *LNCS*. Springer, 1998.

[Mon98] Luís Monteiro. Semantic domains based on sets with families of equivalences. *Electronic Notes in Theoretical Computer Science*, 11:73–106, 1998.

[Mon00] Luís Monteiro. Observation systems. In *Coalgebraic Methods in Computer Science, CMCS'2000*, volume 33 of *Electronic Notes in Theoretical Computer Science*, pages 265–279. Elsevier, 2000.

[MP98] Luís Monteiro and António Porto. Entailment-based actions for coordination. *Theoretical Computer Science*, 192:259–286, 1998.

[MPL00a] M.Bernardo, P.Ciancarini, and L.Donatiello. AEMPA: A Process Algebraic Description Language for the Performance Analysis of Software Architectures. In *Proc. 2nd ACM Int. Workshop on Software and Performance (WOSP2000)*, Ottawa (Canada), September 2000.

[MPL00b] M.Bernardo, P.Ciancarini, and L.Donatiello. On the formalization of architectural types with process algebras. In *Proc. ACM/IEEE Int. Conf. on Fundamentals of Software Engineering (FSE)*, San Diego, November 2000.

[MSC99] A. K. Moran, D. Sands, and M. Carlsson. Erratic Fudgets: A semantic theory for an embedded coordination language. In *Coordination '99*, Lecture Notes in Computer Science. Springer-Verlag, April 1999.

[MT98] U. Montanari and C. Talcott. Can actors and pi-agents live together?,. In Pitts Gordon and Talcott, editors, *Proceedings of Second Workshop on Higher-Order Operational Techniques in Semantics (HOOTS II)*, volume 10 of *ENTCS*. Elsevier, 1998.

[NA00a] Oscar Nierstrasz and Franz Achermann. Separation of concerns through unification of concepts. In *ECOOP 2000 Workshop on Aspects & Dimensions of Concerns*, 2000.

[NA00b] Oscar Nierstrasz and Franz Achermann. Supporting Compositional Styles for Software Evolution. In *Proceedings ISPSE 2000*. IEEE, 2000.

[NFP00] R. De Nicola, G. Ferrari, and R. Pugliese. Programming acess control: The KLAIM experience. In *Proc. CONCUR'2000*, volume 1877 of *Lecture Notes in Computer Science*, 2000.

[NRH99] Flemming Nielson, Hanne Riis Nielson, and Chris Hankin. *Principles of Program Analysis*. Springer, 1999.

[NSA00] Oscar Nierstrasz, Jean-Guy Schneider, and Franz Achermann. Agents everywhere, all the time. In *ECOOP 2000 Workshops on Component-Oriented Programming and Pervasive Component Systems*, 2000.

[OPT00] Karol Ostrovský, K. V. S. Prasad, and Walid Taha. Towards a primitive theory of broadcasting (extended version), October 2000. Available electronically: `http://www.cs.chalmers.se/~karol/Papers/`.

[OTZ00] Andrea Omicini, Robert Tolksdorf, and Franco Zambonelli, editors. *Engineering Societies in the Agent World First International Workshop, ESAW 2000, Berlin, Germany, August 21, 2000*, volume 1972 of *Lecture Notes in Artificial Intelligence*. Springer, 2000.

[PA98] G. A. Papadopoulos and F. Arbab. Modelling electronic commerce activities using control-driven coordination. In *Proceedings of the* 9^{th} *International Workshop on Database and Expert Systems Applications (DEXA 98)*, pages 529–535. IEEE Computer Society, 1998.

[PA00a] G. A. Papadopoulos and F. Arbab. Coordinating electronic commerce activities in Manifold. *Netnomics*, 2(2):101–116, 2000.

[PA00b] G. A. Papadopoulos and F. Arbab. Dynamic reconfiguration in coordination languages. In *Proc. 8th Intl. Conf. on High Performance Computing and Networking, HPCN Europe 2000*, volume 1823 of *Lecture Notes in Computer Science*, pages 197–206. Springer, 2000.

[Paw00] Michel Pawlak. L'implantation d'un espace d'informations distribue et securise. Technical report, University of Geneva, Centre Universitaire d'Informatique, November 6, 2000.

[PR00] António Porto and Gruia-Catalin Roman, editors. *Coordination Languages and Models, 4th International Conference*, volume 1906 of *Lecture Notes in Computer Science*. Springer, 2000.

[Qui99] Francisco Hernández Quiroz. *A Semantics-based Proof System for Gamma*. PhD thesis, Imperial College, 1999.

[RB00] Chrislain Razafimahefa and Ciarán Bryce. Towards the design of an internet operating system. Internet objects, Centre Universitaire d'Informatique, University of Geneva, September 2000.

[Rut98] J.J.M.M. Rutten. Automata and coinduction (an exercise in coalgebra). In *Proceedings of CONCUR'98*, volume 1466 of *Lecture Notes in Computer Science*. Springer, 1998.

[Rut99] J.J.M.M. Rutten. Automata, power series, and coinduction: taking input derivatives seriously. In *Proceedings of ICALP'99*, volume 1644 of *Lecture Notes in Computer Science*, pages 645–654. Springer, 1999. (extended abstract).

[Rut00a] J.J.M.M. Rutten. Behavioural differential equations: a coinductive calculus of streams, automata, and power series. Technical report SEN-R0023, CWI, Amsterdam, 2000.

[Rut00b] J.J.M.M. Rutten. Coalgebra, concurrency, and control. In R. Boel and G. Stremersch, editors, *Discrete Event Systems (analysis and control), Proc. WODES 2000 (5th Workshop on Discrete Event Systems)*, pages 31–38. Kluwer, 2000.

[Rut00c] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.

[RV99] D. Rossi and F. Vitali. Internet-based coordination environemnts and Document-based applications: A case study. In P. Ciancarini and A. Wolf, editors, *Proc. 3rd Int. Conf. on Coordination Models and Languages*, volume 1594 of *Lecture Notes in Computer Science*, pages 259–274, Amsterdam, Netherland, April 1999. Springer.

[SC00a] João Seco and Luís Caires. Parametrically typed components. In *Proceedings of WCOP'00, Workshop on Component Oriented Programming.* University of Karlskrona/Ronneby, 2000. Technical report.

[SC00b] João Seco and Luís Caires. A simple model of typed components. In *Proc. ECOOP'00, European Conference on Object-Oriented Programming*, volume 1850 of *Lecture Notes in Computer Science*, pages 42–56. Springer, 2000.

[Scu99] Adriano Scutellà. Simulation of conference management using an event-driven coordination language. In P. Ciancarini and A. Wolf, editors, *Coordination Languages and Models (Proceedings of the 3rd international conference)*, volume 1594 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 1999.

[SL97] Jean-Guy Schneider and Markus Lumpe. Synchronizing concurrent objects in the pi-calculus. In Roland Ducournau and Serge Garlatti, editors, *Proc. Langages et Modèles à Objets*, pages 61–76. Hermes, 1997.

[SM99] L. Semini and C. Montangero. A refinement calculus for tuple spaces. *Science of Computer Programming*, 34, 1999.

[SN99] Jean-Guy Schneider and Oscar Nierstrasz. Components, scripts and glue. In Leonor Barroca, Jon Hall, and Patrick Hall, editors, *Software Architectures – Advances and Applications*, pages 13–25. Springer, 1999.

[SS99] A. Sabelfeld and D. Sands. A per model of secure information flow in sequential programs. In *Programming Languages and Systems, 8th European Symposium on Programming, ESOP'99*, volume 1575 of *Lecture Notes in Computer Science*, pages 40–58. Springer-Verlag, 1999.

[SS00] A. Sabelfeld and D. Sands. Probabilistic noninterference for multi-threaded programs. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop*, pages 200–214, Cambridge, England, July 2000. IEEE Computer Society Press.

[TBLF00] Robert Tolksdorf, Kirtie Bellman, Chris Landauer, and Alois Ferscha. Final summay report. In *Proceedings of the IEEE Ninth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises WET ICE 2000.* IEEE Computer Society Press, 2000.

[TCD00] Sander Tichelaar, Juan Carlos Cruz, and Serge Demeyer. Design guidelines for coordination components. In Janice Carroll, Ernesto Damiani, Hisham Haddad, and Dave Oppenheim, editors, *Proceedings ACM SAC 2000*, pages 270–277. ACM, March 2000.

[TCG$^+$98] Robert Tolksdorf, Paolo Ciancarini, Mark Ginsburg, Jakob Hummes, and Wilfred Jamison. Working group report on coordination architectures for distributed web applications. In *Proc. 7th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '98).* IEEE Computer Society Press, 1998.

[tHBGK00] P.J. 't Hoen, G. Busatto, L.P.J. Groenwegen, and J.N. Kok. From oo to components: Components at the type and instance level. In *Proceedings SCI*, 2000.

[Tol97] Robert Tolksdorf. Berlinda: An object-oriented platform for implementing coordination languages in java. In David Garlan and Daniel Le Métayer, editors, *Coordination Languages and Models, Proceedings of COORDINATION '97*, LNCS 1282, pages 430–433. Springer, 1997.

[Tol98a] Robert Tolksdorf. Coordination patterns in mobile object spaces. In *Proc. 7th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '98)*. IEEE Computer Society Press, 1998.

[Tol98b] Robert Tolksdorf. Coordination patterns of mobile information agents. In Matthias Klusch and Gerhard Weiß, editors, *Cooperative Information Agents II*, volume 1435 of *Lecture Notes in Artificial Intelligence*, pages 246–261. Springer, 1998.

[Tol98c] Robert Tolksdorf. A framework for multiple coordination languages. *World Wide Web*, 1(2):101–108, 1998.

[Tol98d] Robert Tolksdorf. Laura - a service-based coordination language. *Science of Computer Programming*, 31:359–381, 1998. Special issue on Coordination Models, Languages, and Applications.

[Tol99a] Robert Tolksdorf. On coordinating information agents and mobility. In M. Klusch, editor, *Intelligent Information Agents*, pages 396–411. Springer, 1999.

[Tol99b] Robert Tolksdorf. XML und darauf basierende standards: Die neuen auszeichnungssprachen des Web. *Informatik Spektrum*, 22(6):407–421, 1999.

[Tol00a] Robert Tolksdorf. Coordinating work on the web with workspaces. In *Proceedings of the IEEE Ninth International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises WET ICE 2000*. IEEE Computer Society, Press, 2000.

[Tol00b] Robert Tolksdorf. Coordination technology for workflows on the web: Workspaces. In *Proceedings of the Fourth International Conference on Coordination Models and Languages COORDINATION 2000*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[Tol00c] Robert Tolksdorf. Models of coordination. In Andrea Omicini, Robert Tolksdorf, and Franco Zambonelli, editors, *Engineering Societies in the Agent World First International Workshop, ESAW 2000, Berlin, Germany, August 21, 2000*, volume 1972 of *Lecture Notes in Artificial Intelligence*, pages 78–92. Springer, 2000.

[VBB98] J. Vitek, C. Bryce, and W. Binder. Designing JavaSeal or how to make java sage for agents. In D. Tsichritzis, editor, *Electronic Commerce Objects*. University of Geneva, July 1998.

[VBO01] J. Vitek, C. Bryce, and M. Oriol. Coordinating Processes Using Secure Processes. *Science of Computer Programming*, 2001.

[Vit99] J. Vitek. *The Seal Model of Mobile Computations*. PhD thesis, University of Geneva, March 1999.

[Wei99a] M. Weichert. Pipelining the molecule soup: A plumber's approach to gamma. In *Coordination '99*, Lecture Notes in Computer Science. Springer-Verlag, April 1999.

[Wei99b] Martin Weichert. *Communication and Coordination: On Some Aspects of Broadcasting, Sequential Value Passing and Multiset Transformation*. PhD thesis, Dept. of Computer-Science, Chalmers University, Gothenburg, 1999.

[Zav98] G. Zavattaro. Towards a hierarchy of negative test operators for generative communication. *Electronic Notes in Theoretical Computer Science*, 16(2), 1998. Proc. Fifth Workshop on Expressiveness in Concurrency, EXPRESS'98.