



# Rewriting logic: roadmap and bibliography<sup>☆</sup>

Narciso Martí-Oliet<sup>a,\*</sup>, José Meseguer<sup>b</sup>

<sup>a</sup>*Facultad de Matemáticas, Universidad Complutense, Madrid, Spain*

<sup>b</sup>*Computer Science Laboratory, SRI International, 333 Ravenswood Ave., Menlo Park, CA, 94025 USA*

## 1. Introduction

The theory and applications of rewriting logic have been vigorously developed by researchers all over the world during the past eleven years. The attached bibliography includes more than three hundred papers related to rewriting logic that have been published so far. Three international workshops on rewriting logic have been held in the United States, France, and Japan [222,172,139], and a fourth will be held in Italy in 2002. Furthermore, as explained later in this roadmap, several language implementations and a variety of formal tools have also been developed and have been used in a wide range of applications.

Several snapshots of the state of rewriting logic research — some more global in scope, and others restricted to specific areas such as concurrency or object-based systems — have appeared so far [223,227,229,228]. The present survey is another such snapshot, but it is restricted on purpose on two counts: first in its length, which is relatively short; and second in discussing only work within the rewriting logic area. In particular, no attempt has been made to discuss work on related approaches serving as logical or semantic frameworks. In fact, it is not even a detailed survey of work in rewriting logic; instead, as its name suggests, it is a *roadmap* to help somebody interested in this area get the lay of the land, that is, a first general overview of the main concepts, results, and applications in what we think is a promising research area. In particular, the references cited in the roadmap do not try to be exhaustive, but only

<sup>☆</sup> Supported by DARPA through Rome Laboratories Contract F30602-97-C-0312, by Office of Naval Research Contracts N00014-99-C-0198 and N00014-01-1-0837, and by National Science Foundation Grant CCR-9900334.

\* Corresponding author.

E-mail address: narciso@sip.ucm.es (N. Martí-Oliet).

to give some illustrative examples. However, the bibliography itself contains all the relevant references that we are aware of at this time.

## 2. Basic concepts

In rewriting logic [218] the basic axioms are rewrite rules of the form  $t \rightarrow t'$ , with  $t$  and  $t'$  expressions in a given language. There are two complementary readings of a rewrite rule  $t \rightarrow t'$ , one computational, and another logical:

- *computationally*, the rewrite rule  $t \rightarrow t'$  is interpreted as a *local transition* in a concurrent system; that is,  $t$  and  $t'$  describe patterns for *fragments* of the distributed state of a system, and the rule explains how a local concurrent transition can take place in such a system, changing the local state fragment from an instance of the pattern  $t$  to the corresponding instance of the pattern  $t'$ .
- *logically*, the rewrite rule  $t \rightarrow t'$  is interpreted as an *inference rule*, so that we can infer formulas of the form  $t'$  from formulas of the form  $t$ .

The computational and logical viewpoints are not exclusive: they complement each other and are, in some sense, in the eyes of the beholder. For example, a simple rewrite theory whose rewrite rules rewrite ground multisets built out of some constants by means of an associative and commutative multiset union operator, denoted, say, by  $\otimes$ , has an obvious computational reading as a (place/transition) Petri net; and an equally obvious logical reading as a tensor theory in propositional linear logic (for a discussion of these two readings see [211]).

A rewrite theory is a 4-tuple  $\mathcal{R} = (\Sigma, E, L, R)$ , where  $(\Sigma, E)$  is the equational theory *modulo which* we rewrite,  $L$  is a set of labels, and  $R$  is a set of labeled rules.<sup>1</sup> In the case of a Petri net,  $\Sigma$  consists of the binary multiset union operator  $\otimes$  and one constant for each place in the net,  $E$  consists of the associativity and commutativity equations for multiset union,  $L$  is the set of labels of the net's transitions, and  $R$  is the set of transitions. Since we rewrite *modulo* the equations  $E$ , what are really rewritten are *equivalence classes* of terms modulo  $E$ . In the Petri net example this corresponds to the fact that each transition rewrites (a fragment of) the current multiset of places (graphically depicted as a “marking”, with as many “tokens” in a place as its multiplicity) modulo the associativity and commutativity of multiset union.

As a consequence, the relevant *sentences* — that may or may not be provable by the above theory  $\mathcal{R}$  — are sequents of the form  $[t]_E \rightarrow [t']_E$ , where  $t$  and  $t'$  are  $\Sigma$ -terms, possibly involving some variables, and  $[t]_E$  denotes the equivalence class of the term  $t$  modulo the equations  $E$ . The *provable* sentences are exactly those derivable by the following inference rules:<sup>2</sup>

- (1) *Reflexivity*. For each  $[t] \in T_{\Sigma, E}(X)$ ,  $\overline{[t] \rightarrow [t]}$ .

<sup>1</sup> For simplicity we will assume that  $R$  consists of *unconditional* labeled rules of the form  $l : t \rightarrow t'$ , but all we say extends naturally to conditional rules that may contain rewrites in their conditions [218].

<sup>2</sup> For simplicity, we treat here *unsorted* (and unconditional) rewriting logic; but the logic is in fact parameterized by the choice of its underlying equational logic: unsorted, many-sorted, order-sorted, membership, etc.

(2) *Congruence*. For each  $f \in \Sigma_n$ ,  $n \in \mathbb{N}$ ,

$$\frac{[t_1] \rightarrow [t'_1] \quad \dots \quad [t_n] \rightarrow [t'_n]}{[f(t_1, \dots, t_n)] \rightarrow [f(t'_1, \dots, t'_n)]}.$$

(3) *Replacement*. For each rule  $l: [t(x_1, \dots, x_n)] \rightarrow [t'(x_1, \dots, x_n)]$  in  $R$ ,

$$\frac{[w_1] \rightarrow [w'_1] \quad \dots \quad [w_n] \rightarrow [w'_n]}{[t(\bar{w}/\bar{x})] \rightarrow [t'(\bar{w}'/\bar{x})]}.$$

(4) *Transitivity*.

$$\frac{[t_1] \rightarrow [t_2] \quad [t_2] \rightarrow [t_3]}{[t_1] \rightarrow [t_3]}.$$

Computationally, the provable sequents describe all the complex concurrent transitions of the system axiomatized by  $\mathcal{R}$ . Logically, they describe all the possible complex deductions from one formula to another in the logic axiomatized by  $\mathcal{R}$ .

Besides having an inference system, rewriting logic also has a *model theory* with natural computational and logical interpretations. Furthermore, each rewrite theory  $\mathcal{R}$  has an *initial model*  $\mathcal{T}_{\mathcal{R}}$  [218]. The idea is that we can decorate the provable sequents with *proof terms*, indicating how indeed they have been proved. Computationally, a proof term is a description of a, possibly complex, concurrent computation; logically, it is of course a description of a logical deduction. The question is then, when should two such proof terms be considered *equivalent* descriptions of the same computation/deduction? In the model  $\mathcal{T}_{\mathcal{R}}$  this is answered by equating proof terms according to natural equivalence equations [218]. In this way, we obtain a model  $\mathcal{T}_{\mathcal{R}}$  with a category structure, where the objects are  $E$ -equivalence classes of ground  $\Sigma$ -terms, and the arrows are equivalence classes of proof terms. Identities are naturally associated with reflexivity proofs; and arrow compositions correspond to transitivity proofs. The computational and logical interpretations are then obvious, since a category is a structured transition system; and logical systems have been understood as categories since the early work of Lambek on deductive systems. The proof theory and model theory of rewriting logic are related by a *completeness theorem*, stating that a sequent is provable from  $\mathcal{R}$  if and only if it is satisfied in all models of  $\mathcal{R}$  [218].

Yet another very important property of rewriting logic is *reflection* [67,65,70]. Intuitively, a logic is reflective if it can represent its metalevel at the object level in a sound and coherent way. Specifically, rewriting logic can represent its own theories and their deductions by having a finitely presented rewrite theory  $\mathcal{U}$  that is *universal*, in the sense that for any finitely presented rewrite theory  $\mathcal{R}$  (including  $\mathcal{U}$  itself) we have the following equivalence

$$\mathcal{R} \vdash t \rightarrow t' \Leftrightarrow \mathcal{U} \vdash \langle \bar{\mathcal{R}}, \bar{t} \rangle \rightarrow \langle \bar{\mathcal{R}}, \bar{t}' \rangle,$$

where  $\bar{\mathcal{R}}$  and  $\bar{t}$  are terms, of respective sorts *Module* and *Term*, representing  $\mathcal{R}$  and  $t$  as data elements of  $\mathcal{U}$ . Since  $\mathcal{U}$  is representable in itself, we can achieve a “reflective tower” with an arbitrary number of levels of reflection [65,71].

Reflection is a very powerful property: it allows defining rewriting strategies by means of metalevel theories that extend  $\mathcal{U}$  and guide the application of the rules in a given object-level theory  $\mathcal{R}$  [65]; it can be efficiently supported in a language implementation by means of *descent functions* [71]; it can be used to build a variety of theorem proving and theory transformation tools [65,82,83]; it can endow a rewriting logic language with powerful theory composition operations [120,116,118,124]; and it can be used to prove metalogical properties about families of theories in rewriting logic, and about other logics represented in the rewriting logic logical framework [12,84].

How should rewrite theories be executed in practice? First of all, in a general rewrite theory  $\mathcal{R} = (\Sigma, E, L, R)$  the equations  $E$  can be arbitrary, and therefore,  $E$ -equality may be undecidable. Assuming that the equations  $E$  are unconditional, a general solution is to transform  $\mathcal{R}$  into a rewrite theory  $\mathcal{R}^\# = (\Sigma, \emptyset, L \cup L_E, R \cup E \cup E^{-1})$  in which we view the equations  $E$  as rules from left to right ( $E$ ) and from right to left ( $E^{-1}$ ), labeled by appropriate new labels  $L_E$ . In this way, we can reduce the problem of rewriting modulo  $E$  to the problem of standard rewriting, since we have the equivalence

$$\mathcal{R} \vdash [t] \rightarrow [t'] \Leftrightarrow \mathcal{R}^\# \vdash t \rightarrow t'.$$

In actual specification and programming practice we can do much better than this, because the equational theory  $(\Sigma, E)$  is typically decidable. A commonly occurring form for the decidable equational theory  $(\Sigma, E)$  is with  $E = E' \cup A$ , where  $A$  is a set of equational axioms for which we have a matching algorithm, and  $E'$  is a set of Church–Rosser and terminating equations *modulo*  $A$ . In these circumstances, a very attractive possibility is to transform  $\mathcal{R} = (\Sigma, E' \cup A, L, R)$  into the theory  $\mathcal{R}^\dagger = (\Sigma, A, L \cup L_{E'}, R \cup E')$ . That is, we now view the equations  $E'$  as rules added to  $R$ , labeled with appropriate new labels  $L_{E'}$ . In this way, we reduce the problem of rewriting modulo  $E$  to the much simpler problem of rewriting modulo  $A$ , for which, by assumption, we have a matching algorithm. The question is, of course, under which conditions is this transformation complete, that is, under which conditions do we have an equivalence

$$\mathcal{R} \vdash [t]_E \rightarrow [t']_E \Leftrightarrow \mathcal{R}^\dagger \vdash [t]_A \rightarrow [t']_A.$$

Conditions guaranteeing this equivalence center around different variations on the notion of *coherence*, which is a form of “relative confluence” between equations and rules. Methods for checking coherence, or for achieving it by a process of “relative completion”, have been proposed by Viry in several papers [314,315,318].

Even when the rewrite theory is coherent and the language implementation supports rewriting modulo  $A$ , executing rewrite theories is nontrivial, because the rules  $R$  in general are neither Church–Rosser nor terminating. Furthermore, some rules in  $R$  may have *additional variables* on their right-hand sides, yet another source of non-determinism. For this reason, sequential implementations of rewriting logic typically support *rewriting strategies* that let the user specify how the rules should be applied [183,67,23,68,17,319,65]. Such strategies can be defined in metalevel theories by reflection, as already indicated, or they may be part of a strategy language supported by a language implementation. However, one should not forget that rewriting logic is an *intrinsically concurrent formalism*, that can be used directly for concurrent and

distributed programming (see for example [235,203,128]). Therefore, whereas in a sequential implementation we are *simulating* a concurrent execution, and need a strategy to choose a particular interleaving computation, in a truly concurrent execution non-determinism is a fact of life, and we may care much less about how rules are applied, and be much less able to control their application in practice. We may in fact allow many different computations, while still imposing some weaker requirements such as different forms of fairness.

### 3. Rewriting logic and formal methods

The fact that, under reasonable assumptions, rewriting logic specifications are executable allows us to have a flexible range of *increasingly stronger formal methods*, to which a system specification can be subjected. Only after less costly and “lighter” methods have been used, it is meaningful and worthwhile to invest effort on “heavier” and costlier methods. A rewriting logic language implementation, together with an associated environment of formal tools, can be used to support the following, increasingly stronger methods [79]: (1) formal specification, (2) execution of the specification, (3) model-checking analysis, (4) narrowing analysis, and (5) formal proof.

Executability, combined with program transformation and compilation techniques, has yet another key advantage, namely, that rewriting logic specifications validated by the above formal methods can then be directly transformed and compiled for efficient execution. In fact, the state-of-the-art in rewriting logic language implementations (see Section 6) suggests that for many applications the implementations thus obtained, besides being correct by construction, can compete in efficiency with implementations developed in conventional languages.

The above methodology should be supported by formal tools. First of all, a reflective rewriting logic implementation can directly support methods 1–3, and can also be used as a *reflective metatool* to develop other formal tools for methods 3–5. Maude has been used in exactly this way [82,83,262] to build tools such as an inductive theorem prover; a tool to check the Church–Rosser property, coherence, and termination, and to perform Knuth–Bendix completion; and a tool to specify, analyze and model check real-time specifications [265,262]. Some of the above tools have also been integrated within the formal tool environment of CafeOBJ [143]. Similarly, as further discussed in Section 5, both ELAN and Maude have been used to develop a wide variety of formal tools and automated deduction algorithms, based on quite different logics.

Rewriting logic is primarily a logic *of* change in which the deduction directly corresponds to the change [211], as opposed to a logic to talk *about* change in a more indirect and global manner, such as the different variants of modal and temporal logic. Such logics regard a system as a mathematical model — typically some kind of Kripke structure — about which they then make assertions about its global properties, such as safety or liveness properties. Both levels of description and analysis are useful in their own right; in fact, they complement each other: one can use both logics in combination to prove system properties.

The integration of these two logical levels is usually straightforward, because both logics are talking about essentially the same mathematical model. In fact, the initial model  $\mathcal{T}_{\mathcal{R}}$  of a rewrite theory  $\mathcal{R}$  is a category with algebraic structure, where the objects correspond to system states, and the arrows correspond to concurrent system transitions. Therefore,  $\mathcal{T}_{\mathcal{R}}$  can be regarded as a Kripke structure whose transitions are labeled by the arrows of the category. A variety of different modal or temporal logics can then be chosen to make assertions about such a Kripke structure, or about a closely-related structure obtained from it, such as, for example, the extension  $\mathcal{T}_{\mathcal{R}}^{\infty}$  of  $\mathcal{T}_{\mathcal{R}}$  to infinite computations.

The investigation of suitable specification logics having a good integration with rewriting logic is an active area of research. In the choice of such a specification logic there are different tradeoffs between, for example, generality, expressiveness, and amenability to different deductive and/or model-checking techniques. Two general proposals for modal logics for reasoning about general rewrite theories are those of Fiadeiro et al. in [136], and the coalgebraic approach of Pattinson [270]. But since object-oriented systems constitute a particularly wide and important application area, modal or temporal logics that provide explicit support for object systems and can reason about their rewriting logic specifications are clearly of interest. Two candidate formalisms of this kind have been proposed. One is a version of the modal  $\mu$ -calculus proposed by Lechner for reasoning about object-oriented Maude specifications [194,195,198], and another is Denker's object-oriented distributed temporal logic [90]. A direction recently explored by Ölveczky and supported by the model-checking features of the Real-Time Maude tool [265] is a timed linear time temporal logic suitable for reasoning about rewriting logic specifications of real-time systems [262]; in a similar vein, Beffara et al. have used rewrite rules and ELAN strategies to verify properties of timed automata [14]. An even more recent direction actively pursued at SRI is the development of an explicit state model checker to check linear temporal logic formulas on the general class of rewriting logic specifications executable in Maude; this model checker will be part of the upcoming Maude 2.0 distribution.

#### 4. Semantic framework applications

The computational and logical interpretations of rewriting logic lead to applications that use it: as a *semantic framework*, in which different languages and models of computation are expressed; or as a *logical framework*, in which different logics and inference systems are likewise expressed [208]. We first discuss semantic framework applications.

##### 4.1. Models of computation

This section presents concrete evidence (in highly condensed form; see [223,227] for much more detailed discussions) for the thesis that a wide variety of models of computation, including concurrent ones, can be naturally and directly expressed

as rewrite theories in rewriting logic. As a consequence, models hitherto quite different from each other can be naturally unified and interrelated within a common framework.

The following models of computation have been naturally expressed in rewriting logic: (1) equational programming, which is the special case of rewrite theories whose set of rules is empty and whose equations are Church–Rosser, possibly modulo some axioms  $A$ ; (2) lambda calculi and combinatory reduction systems [218,192,193,293,290]; (3) labeled transition systems [218]; (4) grammars and string-rewriting systems [218]; (5) Petri nets, including place/transition nets, contextual nets, algebraic nets, colored nets, and timed Petri nets [218,223,291,295,266,287]; (6) Gamma and the Chemical Abstract Machine [218]; (7) CCS and LOTOS [237,208,314,45,89,311,309,201]; (8) the  $\pi$  calculus [316,290]; (9) concurrent objects and actors [218,220,300,302,304]; (10) the UNITY language [218]; (11) concurrent graph rewriting [223]; (12) dataflow [223]; (13) neural networks [223]; (14) real-time systems, including timed automata, timed transition systems, hybrid automata, and timed Petri nets [266,262]; and (15) the tile logic [146,147,135] model of synchronized concurrent computation [231,40,34,148].

Since the above specifications of models of computation as rewrite theories are typically executable, this suggests that rewriting logic is a very flexible *operational* semantic framework to specify the semantics of such models. What is not immediately apparent from the above list is that it is also a flexible *mathematical* semantic framework at the level of concurrency models. That is, quite often a well-known mathematical model of concurrency happens to be isomorphic to the initial model  $\mathcal{T}_{\mathcal{R}}$  of the rewrite theory  $\mathcal{R}$  axiomatizing that particular model, or at least closely related to such an initial model. Some examples will illustrate this point: (1) in [193] it is shown that for rewrite theories of the form  $\mathcal{R} = (\Sigma, \emptyset, L, R)$ , with the rules  $R$  left-linear,  $\mathcal{T}_{\mathcal{R}}$  is isomorphic to a model based on residuals and permutation equivalence proposed by Boudol; (2) the same paper also shows that for  $\mathcal{R} = (\Sigma, E, L, R)$  a rewrite theory axiomatizing an orthogonal combinatory reduction system, including the  $\lambda$ -calculus, a quotient of  $\mathcal{T}_{\mathcal{R}}$  by a few additional equations is isomorphic to a well-known model of parallel reductions based on residuals and permutation equivalence; (3) the paper [295] shows in detail that for  $\mathcal{R} = (\Sigma, E, L, R)$  a rewrite theory axiomatizing a place/transition net,  $\mathcal{T}_{\mathcal{R}}$  is naturally isomorphic (in the categorical sense) to the Best-Devillers net process model — a result essentially known from the coincidence of  $\mathcal{T}_{\mathcal{R}}$  with the Meseguer–Montanari algebraic model of nets [218] and the Degano–Meseguer–Montanari algebraic characterization of net processes — and then generalizes this natural isomorphism to one between  $\mathcal{T}_{\mathcal{R}}$  and a Best-Devillers-like model for  $\mathcal{R}$  the axiomatization of an algebraic net; (4) the papers [45,89] show that for  $\mathcal{R} = (\Sigma, E, L, R)$  a rewrite theory axiomatizing CCS, a truly concurrent semantics causal model based on proved transition systems is isomorphic to a quotient of  $\mathcal{T}_{\mathcal{R}}$  by a few additional axioms; (5) the paper [237] shows that for  $\mathcal{R} = (\Sigma, E, L, R)$  a rewrite theory axiomatizing a concurrent object-oriented system satisfying reasonable requirements, a subcategory of  $\mathcal{T}_{\mathcal{R}}$  is isomorphic to a partial order of events model which, for asynchronous object systems corresponding to actors, coincides with the finitary part of the Hewitt–Baker partial order of events model.



An important additional development in this area is the  $\rho$ -calculus of Cirstea and Kirchner [57,54,59,60]. This is a very general rewrite theory that can play for rewriting logic specifications a role similar to that played by the  $\lambda$ -calculus in functional computing; its generality is shown by the fact that  $\rho$ -terms generalize the rewriting logic proof terms defined in [218]. Furthermore, the  $\rho$ -calculus can simulate the  $\lambda$ -calculus itself. In fact, by replacing and generalizing the  $\lambda$ -calculus idea of function application by that of *rule application*, the  $\rho$ -calculus unifies both the  $\lambda$ -calculus and first-order rewriting. In analogy with  $\lambda$ -calculi, there are typed versions, including a simply typed  $\rho$ -calculus and a “ $\rho$  cube” [58,62].

#### 4.2. Semantics of programming languages

Rewriting logic is a promising semantic framework for formally specifying programming languages as rewrite theories. Since those specifications usually can be executed in a rewriting logic language, they in fact become *interpreters* for the languages in question. In addition, such formal specifications allow both formal reasoning and a variety of formal analyses for the languages so specified.

The use of rewrite rules to define the semantics of programming languages is of course not new. In a higher-order version it goes back to the use of semantic equations in denotational semantics; in a first-order version, the power of equational specifications to give semantic definitions of conventional languages has been understood and used for a long time. However, both the lambda calculus and executable equational specifications implicitly assume that such language definitions can be given in terms of *functions*, and rely on the Church–Rosser property to reach the result of an execution. For sequential languages, by making the state of the computation explicit, a functional description of this kind can always be achieved. The situation becomes more difficult for languages that support highly concurrent and nondeterministic applications, and where the possibly nonterminating *interactions* between processes or components — as opposed to the computation of an output value from given inputs — are often the whole point of a program. Such languages and applications do not have a natural equational description in terms of functions, but do have a very natural rewriting logic semantics, not only operationally (by means of rewriting steps) but also denotationally ( $\mathcal{T}_A$  and related models).

Since structural operational semantics definitions can be used for languages not amenable to a functional description, it is natural to compare them with rewriting logic definitions. Their relationship has been discussed in detail in [208]. In fact, both “big-step” and “small-step” structural operational semantics definitions can be naturally regarded as special formats of corresponding rewrite theory definitions [208]. Tile models provide yet another systematic way of understanding structural operational semantics definitions as tile rewrite theories [146–148], which can then be mapped into rewriting logic for execution purposes [231,40,34]. There is also a close connection between rewriting logic and Mosses’s modular structural operational semantics (MSOS) which has been recognized from the beginning [247,248], and that has led to ongoing work on a Maude Action Tool to execute MSOS definitions and Action Semantics definitions [32].



A number of encouraging case studies giving rewriting logic definitions of programming languages have already been carried out by different authors. Firstly, some of the models of computation discussed in Section 4.1 are so closely connected with languages that their rewriting logic specifications are also language specifications. Good examples are rewriting logic definitions of the lambda calculus and (mini-) ML, CCS, the  $\pi$ -calculus, and sketches of UNITY and Gamma, which are given in some of the references cited in Section 4.1. Secondly, the usefulness of rewriting strategies to specify program evaluations has been clearly demonstrated in ELAN specifications, for example of Prolog and of the functional-logic programming language Babel [320], and also in the Maude executable specifications for CCS developed by Bruni and Clavel [63,34], and by Verdejo and Martí-Oliet [311,310]. Thirdly, the fact that rewriting logic naturally supports concurrent objects has proved very useful in formally specifying a number of novel concurrent and mobile languages. For example, Ishikawa et al. have given a Maude specification of a representative subset of GAEA, a reflective concurrent logic programming language developed at ETL, Japan [164,163]. Mason and Talcott have used rewriting logic to give semantic definitions of actor languages, and to “compile away” certain language features by defining semantics-preserving translations between actor languages that are formalized as translations between their corresponding rewrite theories [212]. Van Baalen, Caldwell, and Mishra have used Maude to give a formal semantics to the DaAgent mobile agent system and to analyze key fault-tolerant protocols in that language [9]; their analysis has revealed mistakes and inconsistencies in the protocols’ informal specifications. Yet another example is the formal executable specification in Maude of UPenn’s PLAN active network programming language [236,322]. Maude itself has been used to define the semantics of its Mobile Maude extension [120]. Finally, Maude has been used not only to specify programming languages, but also to specify and verify microprocessors in work by Harman [154,155].

#### 4.3. Distributed architectures and components

It is very important to detect errors and inconsistencies as early as possible in the software design cycle. For this reason, formal approaches that can increase the analytic power of architectural notations such as architectural description languages (ADLs) and object-oriented design formalisms like UML are quite valuable. A related concern is the formal specification and analysis of *distributed component architectures*.

Rewriting logic has been used by several authors in these areas to allow formal analysis of software designs and, in some cases, to support code generation from the associated executable specifications. Relevant work in this direction includes: (1) work of Nodelman and Talcott representing both the Wright architecture description language and its underlying CSP semantics in Maude; (2) work of Durán, Meseguer, and Talcott on semantic interoperation of heterogeneous software architectures based on their rewriting logic semantics [232] (see also Appendix E of [74]); (3) work of Wirsing and Knapp on the systematic transformation of UML diagrams and similar object-oriented notations into formal executable rewriting logic specifications in Maude, which can

then be used to execute and formally analyze the designs, and even to generate code in a conventional language such as Java [326,185,186,327]; (4) work by Fernández and Toval formalizing in Maude the UML metamodel and its evolution [305,132], with applications to formal analysis and prototyping [131,306]; (5) work by Nakajima and Futatsugi on the transformation of GILO-2 scenario-based object-oriented design diagrams for execution and formal analysis [254]; (6) work by Talcott on a rewriting logic semantics for actor systems axiomatized by actor theories [298–302]; such systems can be extended by an algebra of *components*, that are encapsulated by interfaces, and that can include actors, messages, and other (sub-)components; in addition Talcott has developed methods to reason formally about such open component systems; (7) work by Denker, Meseguer, and Talcott on a general middleware architecture for composable distributed communication services such as fault-tolerance, security, and so on, that can be composed and can be dynamically added to selected subsets of a distributed communications system [100]; (8) work by Najm and Stefani giving a rewriting logic semantics to the operational subset of the Reference Model for Open Distributed Processing (RM-ODP) [249–251] (see also [127]); (9) work by Nakajima that uses rewriting logic specifications in CafeOBJ to formally specify the architecture of WEB-NMS, a Java/ORB implementation of a network management system [252]; and (10) work by Albarrán, Durán, and Vallecillo on interoperating Maude executable specifications with distributed component platforms such as CORBA and SOAP [1–3].

#### 4.4. Specification and analysis of communication protocols

Because of its flexibility to model distributed objects with different modes of communication and interaction, rewriting logic is very well suited to specify and analyze communication protocols, including cryptographic protocols, and, more generally, network software such as active network programming languages, active network algorithms, and network management systems.

Applications of this kind include: (1) work by researchers at Stanford, SRI, and at the Computer Communications Research Group at University of California Santa Cruz using Maude to analyze the early design of a new reliable broadcast protocol for active networks [96,97]; (2) work of Denker, Meseguer, and Talcott on the specification and analysis of cryptographic protocols using Maude [98,99] (see also [277]); (3) work of Basin and Denker on an experimental comparison of the advantages and disadvantages of using Maude versus using Haskell to analyze security protocols [10]; (4) work of Millen and Denker at SRI using Maude to give a formal semantics to their new cryptographic protocol specification language CAPSL, and to endow CAPSL with an execution and formal analysis environment [92–95]; (5) work of Wang, Gunter, and Meseguer using Maude to formally specify and analyze a PLAN active network algorithm [322]; (6) work by Ölveky et al. using Real-Time Maude to specify and analyze the AER/NCA suite of active network protocol components for reliable multicast [268]; (7) work of Verdejo, Pita, and Martí-Oliet on the Maude specification and verification of the FireWire leader election protocol [312]; (8) work of Mason and Talcott on modeling, simulation and analysis of network architectures and communication protocols [213]; and (9) work of Pita and Martí-Oliet using the reflective features of

Maude to specify some management processes of broadband telecommunication networks [271–273].

## 5. Logical framework applications

Rewriting logic is like a coin with two inseparable sides: one computational and another logical. The generality and expressiveness of rewriting logic as a semantic framework for concurrent computation has also a logical counterpart. Indeed, rewriting logic is also a promising *logical framework* in which many different logics and formal systems can be naturally represented and interrelated [208,209]. Using a rewriting logic implementation such representations can then be used to generate a wide range of formal tools.

### 5.1. Representing, mapping, and reasoning about logics

The basic idea is that we can represent a logic  $\mathcal{L}$  with a finitary syntax and inference system within rewriting logic by means of a representation map

$$\Phi: \mathcal{L} \rightarrow RWLogic.$$

The map  $\Phi$  should be *conservative*, that is, it should preserve and reflect theoremhood. The reason why rewriting logic is a good framework is that the formulas of a logic  $\mathcal{L}$  can typically be equationally axiomatized by an equational theory, and the rules of inference can then be typically understood as rewrite rules, that may be conditional if the inference rules have “side conditions”. Therefore, the mappings  $\Phi$  are usually very simple and direct. Furthermore, using reflection we can define and execute a map  $\Phi$  of this kind *inside rewriting logic itself* by means of an equationally defined map

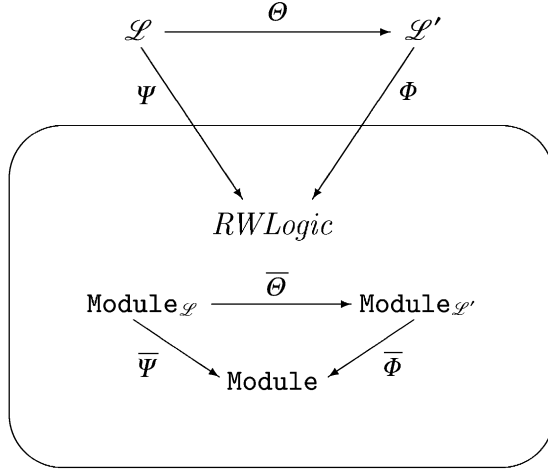
$$\bar{\Phi}: \text{Module}_{\mathcal{L}} \rightarrow \text{Module}.$$

The map  $\bar{\Phi}$  can be defined by extending the universal theory  $\mathcal{U}$ , which has a sort `Module` representing rewrite theories (see Section 2), with the equational definition of a new sort `Module $\mathcal{L}$`  whose terms represent (finitely presentable) theories in the logic  $\mathcal{L}$ .

In fact, we can go a step further, and represent inside rewriting logic a mapping  $\Theta: \mathcal{L} \rightarrow \mathcal{L}'$  between any two finitary logics  $\mathcal{L}$  and  $\mathcal{L}'$  as an equationally defined function  $\bar{\Theta}: \text{Module}_{\mathcal{L}} \rightarrow \text{Module}_{\mathcal{L}'}$ . If the map  $\Theta$  is computable, then, by a metatheorem of Bergstra and Tucker it is possible to define the function  $\bar{\Theta}$  by means of a finite set of Church–Rosser and terminating equations. That is, such functions can be effectively defined and executed within rewriting logic.

In summary, using reflection, mappings between logics, including maps representing other logics in rewriting logic, can be internalized and executed within rewriting logic,

as indicated in the picture below.



There is yet another reason why rewriting logic is very useful for logical framework applications. Thanks to reflection and the existence of initial models, rewriting logic can not only be used as a logical framework in which the deduction of a logic  $\mathcal{L}$  can be faithfully simulated, but also as a *metalogical framework* in which we can reason about the metalogical properties of a logic  $\mathcal{L}$ . Basin, Clavel, and Meseguer have begun studying the use of reflection, induction, and Maude’s inductive theorem prover enriched with reflective reasoning principles to prove such metalogical properties [11–13].

A good number of examples of representations of logics in rewriting logic have been given by different authors, often in the form of executable specifications, including: (1) the logics represented by Martí-Oliet and Meseguer in [208,209], including equational logic, Horn logic with equality, linear logic, logics with quantifiers, and any sequent calculus presentation of a logic for a very general notion of “sequent”; (2) the map  $LinLogic \rightarrow RWLogic$  in [208,209] representing propositional linear logic was subsequently specified in a reflective way in Maude by Clavel and Martí-Oliet [63,65]; (3) the map  $HOL \rightarrow Nuprl$  between the logics of the HOL and Nuprl theorem provers has been specified in Maude by Stehr, Naumov, and Meseguer [257,296]; (4) Dowek, Hardin, and Kirchner have presented (what obviously are) rewrite theories for doing deduction *modulo* an equational theory of equivalence between formulas specified by the equations  $E$  of the rewriting logic axiomatization, both for first-order and higher-order logics [109–111]; (5) the connections with rewriting logic of that work have been made explicit by Viry, who has given a coherent sequent calculus rewrite theory in this style in [317,318] (see also [101]); (6) Stehr and Meseguer have defined a natural representation map  $PTS \rightarrow RWLogic$  of pure type systems (a parametric family of higher-order logics generalizing the  $\lambda$ -cube) in rewriting logic [293]; and (7) Bruni, Meseguer, and Montanari have defined a mapping  $Tile\ Logic \rightarrow RWLogic$  from tile logic into rewriting logic that can be used to execute tile logic specifications [34,37–40].

## 5.2. Specifying and building formal tools

Theorem provers and other formal tools have underlying *inference systems* that can be naturally specified and prototyped in rewriting logic. Furthermore, the strategy aspects of such tools and inference systems can then be specified by rewriting strategies. The researchers in the ELAN group have developed an impressive collection of rewriting logic specifications for different automated deduction inference systems, including the already-mentioned theorem proving modulo methods [109–111], logical languages, unification and narrowing [183,320], Knuth–Bendix completion with constraints [173], higher-order unification [15], combination of unification algorithms [275], constraint solving [46–50], and termination and tree-automata techniques [149,150]. In a somewhat similar vein, the work of Schorlemmer explores the relationships between rewriting logic and Levy and Agustí’s general bi-rewriting approach to automated deduction [282–285].

In Maude, formal tools have typically a reflective design that, by metarepresenting theories as data, easily allows inference steps that may transform the object theory. Strategies are then rewrite theories controlling the application of such metalevel inference rules at the meta-metalevel. We have already mentioned in Section 3 several such tools that are part of the Maude formal environment, namely, an inductive theorem prover; Church–Rosser, coherence, and termination checkers, and a Knuth–Bendix completion tool [80–83,117,119,123]; plus the Real-Time Maude tool [265,262,267]. Also closely related to Maude itself is the Full Maude tool, which extends Maude with special syntax for object-oriented specifications, and with a rich *module algebra* of parameterized modules and module composition operations [120,116,126]. This method of building formal tools is not restricted to Maude-related tools: One can generate tools from their rewriting logic specifications for *any* finitary logic, such as: (1) a proof assistant built by Stehr for the Open Calculus of Constructions, which extends Coquand and Huet’s calculus of constructions with equational reasoning and a flexible universe hierarchy [292]; (2) the Maude Action Tool [32] already mentioned in Section 4.2; (3) a CCS execution and verification environment developed by Verdejo and Martí-Oliet [311,310]; (4) a tool by Havelund and Roşu for testing linear temporal logic formulae on finite execution traces [157–160,278]; and (5) a tool by Fischer and Roşu to automatically check an abstract interpretation against user-given properties [137].

## 6. Language implementations

Several language implementation efforts in France, Japan, and the US have adopted rewriting logic as their semantic basis and support executable rewriting logic specification and programming.

The ELAN language has been developed at LORIA (CNRS, INRIA, and Universities of Nancy) [183,320,26–28,20]. It has as modules *computational systems*, consisting of a rewrite theory and a *strategy* to guide the rewriting process [23,219,17,29]. As already discussed in Section 5, this group and their collaborators have developed a very impressive collection of examples and case studies in areas such as logic programming languages, constraint solving, higher-order unification, equational theorem-proving, and

other such computational systems. Besides the ELAN interpreter, there is also a high-performance ELAN compiler, including compilation of AC-rewriting [243–246,176].

The CafeOBJ language implementation, developed at the Japan Advanced Institute of Science and Technology (JAIST) in Kanazawa [141,140,104,105,108], contains OBJ as its functional sublanguage, and supports object-oriented specifications. Furthermore, its semantics is multi-logical and includes hidden-sorted versions of equational and rewriting logic [102–105]. The CafeOBJ language has been the basis of an ambitious research effort — the Cafe Project — involving several research institutions in Japan, Europe and the US, as well as several Japanese industries, to promote formal methods applications in software engineering [138,143]. This project has achieved a distributable version of the language and further work on its semantics, a collection of specification libraries and case studies, an environment, and a collection of theorem proving tools supporting different forms of verification. Furthermore, a compiler has been developed in addition to the Cafe interpreter implementation [260,165].

The Maude language has been developed at SRI, in Menlo Park, California [220,85,64,79,76]. The equational logic underlying Maude’s rewriting logic is membership equational logic [226,30,31], and gives rise to a sublanguage of *functional modules*. *System modules* specify general rewrite theories, and *object-oriented modules* provide syntactic sugar for object-oriented rewrite theories. These modules can be combined by module composition operations supported by Full Maude [116,126,121]. Maude’s high-performance rewrite engine makes extensive use of advanced semicompile techniques; there is also a high-performance experimental Maude compiler. In addition, Maude efficiently supports reflection through its META-LEVEL module [71,79]. Maude has been used in a wide range of applications, many of which have been discussed in this paper.

## Acknowledgements

We would like to thank all our colleagues who have given us comments on earlier versions of this roadmap and have helped us in compiling the following bibliography; we thank specially Miguel Palomino, whose extensive help has made possible the existence of the bibliography.

## References

- [1] A. Albarrán, F. Durán, A. Vallecillo, From Maude specifications to SOAP distributed implementations: a smooth transition, In: Proc. VI Jornadas de Ingeniería del Software y Bases de Datos, JISBD 2001, Almagro, Spain, November 21–23, 2001.
- [2] A. Albarrán, F. Durán, A. Vallecillo, Maude meets CORBA, in: Proc. 2nd Argentine Symp. on Software Engineering, Buenos Aires, Argentina, September 10–11, 2001.
- [3] A. Albarrán, F. Durán, A. Vallecillo, On the smooth implementation of component-based system specifications, in: Proc. 6th ECOOP Internat. Workshop on Component-Oriented Programming, WCOP’01, Budapest, Hungary, June, 2001.
- [4] N. Aoumeur, G. Saake, On the specification and validation of cooperative information systems using an extended Maude, in: K. Futatsugi, J.A. Goguen, J. Meseguer (Eds.), OBJ/CafeOBJ/Maude Workshop

- at Formal Methods '99: Formal Specification, Proof, and Applications, Theta, Bucharest, 1999, pp. 95–114.
- [5] E. Astesiano, G. Reggio, Algebraic specification of concurrency, in: M. Bidoit, C. Choppy (Eds.), Recent Trends in Data Type Specification, 8th Workshop on Specification of Abstract Data Types Joint with the 3rd COMPASS Workshop, Dourdan, France, August 26–30, 1991, Selected Papers, Lecture Notes in Computer Science, Vol. 655, Springer, Berlin, 1993, pp. 1–39.
  - [6] E. Astesiano, G. Reggio, Labelled transition logic: an outline, Technical Report DISI-TR-96-20, Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova, Italy, 1996.
  - [7] E. Astesiano, G. Reggio, On the relationship between labelled transition logic and rewriting logic, Technical Report DISI-TR-97-23, Dipartimento di Informatica e Scienze dell'Informazione, Università di Genova, Italy, 1997.
  - [8] E. Astesiano, M. Broy, G. Reggio, Algebraic specification of concurrent systems, in: E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner (Eds.), Algebraic Foundations of Systems Specification, IFIP State-of-the-art Reports, Springer, Berlin, 1999, pp. 467–520.
  - [9] J. Van Baalen, J.L. Caldwell, S. Mishra, Specifying and checking fault-tolerant agent-based protocols using Maude, in: First Goddard Workshop on Formal Approaches to Agent-Based Systems, Greenbelt, MD, USA, April 5–7, 2000, Proceedings, Lecture Notes in Computer Science, Springer, Berlin, to appear.
  - [10] D. Basin, G. Denker, Maude versus Haskell: an experimental comparison in security protocol analysis, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 235–256. <http://www.elsevier.nl/locate/entcs/volume36.html>.
  - [11] D. Basin, M. Clavel, J. Meseguer, Reflective metalogical frameworks, in: Proc. of LFM'99: Workshop on Logical Frameworks and Meta-languages, Paris, France, September 28, 1999, <http://www.cs.bell-labs.com/~felty/LFM99/>.
  - [12] D. Basin, M. Clavel, J. Meseguer, Rewriting logic as a metalogical framework, in: S. Kapoor, S. Prasad (Eds.), 20th Conf. on the Foundations of Software Technology and Theoretical Computer Science, New Delhi, India, December 13–15, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1974, Springer, Berlin, 2000, pp. 55–80.
  - [13] D. Basin, M. Clavel, J. Meseguer, Rewriting logic as a metalogical framework, Manuscript, Computer Science Laboratory, SRI International, 2000, submitted for publication.
  - [14] E. Boffara, O. Bournez, H. Kacem, C. Kirchner, Verification of timed automata using rewrite rules and strategies, in: N. Dershowitz, A. Frank (Eds.), Proc. BISFAI 2001, 7th Biennial Bar-Ilan International Symposium on the Foundations of Artificial Intelligence, Ramat-Gan, Israel, June 25–27, 2001.
  - [15] P. Borovanský, Implementation of higher-order unification based on calculus of explicit substitutions, in: M. Bartosek, J. Staudek, J. Wiedermann (Eds.), SOFSEM'95: Theory and Practice of Informatics, 22nd Seminar on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic, November 23–December 1, 1995, Proceedings, Lecture Notes in Computer Science, Vol. 1012, Springer, Berlin, 1995, pp. 363–368.
  - [16] P. Borovanský, Controlling rewriting: study and implementation of a strategy formalism, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 63–74. <http://www.elsevier.nl/locate/entcs/volume15.html>.
  - [17] P. Borovanský, Le Contrôle de la Réécriture: Étude et Implantation d'un Formalisme de Stratégies, Ph.D. Thesis, Université Henri Poincaré — Nancy I, October, 1998.
  - [18] P. Borovanský, C. Castro, Cooperation of constraint solvers: using the new process control facilities of ELAN, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 379–398, <http://www.elsevier.nl/locate/entcs/volume15.html>.
  - [19] P. Borovanský, H. Kirchner, Strategies of ELAN: meta-interpretation and partial evaluation, in: M.P.A. Sellink (Ed.), 2nd Internat. Workshop on the Theory and Practice of Algebraic Specifications,



- Amsterdam, The Netherlands, September 25–26, 1997, Electronic Workshops in Computing. Springer, Berlin, 1998, <http://www.ewic.org.uk/ewic/workshop/view.cfm/ASFSDf-97>.
- [20] P. Borovanský, H. Cirstea, H. Dubois, C. Kirchner, H. Kirchner, P.-E. Moreau, C. Ringeissen, M. Vittek, ELAN v 3.3 user manual, 3rd ed., Technical report, INRIA Lorraine & LORIA, Nancy, France, December, 1998.
- [21] P. Borovanský, S. Jamoussi, P.-E. Moreau, C. Ringeissen, Handling ELAN rewrite programs via an exchange format, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 207–224, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [22] P. Borovanský, C. Kirchner, H. Kirchner, Controlling rewriting by rewriting, in: J. Meseguer, (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 168–188, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [23] P. Borovanský, C. Kirchner, H. Kirchner, Strategies and rewriting in ELAN, in: B. Gramlich, H. Kirchner, (Eds.), Proc. of the CADE-14 Workshop on Strategies in Automated Deduction, Townsville, Australia, July, 1997, pp. 13–24.
- [24] P. Borovanský, C. Kirchner, H. Kirchner, A functional view of rewriting and strategies for a semantics of ELAN, in: M. Sato, Y. Toyama (Eds.), The 3rd Fuji Internat. Symp. on Functional and Logic Programming, Kyoto, Japan, World Scientific, Singapore, 1998, pp. 143–167.
- [25] P. Borovanský, C. Kirchner, H. Kirchner, Rewriting as a unified specification tool for logic and control: the ELAN language, in: M.P.A. Sellink (Ed.), 2nd Internat. Workshop on the Theory and Practice of Algebraic Specifications, Amsterdam, The Netherlands, September 25–26, 1997, Electronic Workshops in Computing, Springer, Berlin, 1998, <http://www.ewic.org.uk/ewic/workshop/view.cfm/ASFSDf-97>.
- [26] P. Borovanský, C. Kirchner, H. Kirchner, P.-E. Moreau, ELAN from the rewriting logic point of view, Theoret. Comput. Sci., 2002, submitted for publication.
- [27] P. Borovanský, C. Kirchner, H. Kirchner, P.-E. Moreau, C. Ringeissen, An overview of ELAN, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 329–344, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [28] P. Borovanský, C. Kirchner, H. Kirchner, P.-E. Moreau, M. Vittek, ELAN: a logical framework based on computational systems, in: J. Meseguer, (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 35–50, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [29] P. Borovanský, C. Kirchner, H. Kirchner, C. Ringeissen, Rewriting with strategies in ELAN: a functional semantics, Internat. J. Foundations Comput. Sci. 12 (2001) 69–95.
- [30] A. Bouhoula, J.-P. Jouannaud, J. Meseguer, Specification and proof in membership equational logic, in: M. Bidoit, M. Dauchet (Eds.), TAPSOFT'97: Theory and Practice of Software Development, 7th Internat. Joint Conf. CAAP/FASE, Lille, France, April 14–18, 1997, Proceedings, Lecture Notes in Computer Science, Vol. 1214, Springer, Berlin, 1997, pp. 67–92.
- [31] A. Bouhoula, J.-P. Jouannaud, J. Meseguer, Specification and proof in membership equational logic, Theoret. Comput. Sci. 236 (2000) 35–132.
- [32] C. Braga, H. Haeusler, J. Meseguer, P. Mosses, Maude action tool: using reflection to map action semantics to rewriting logic, in: T. Rus (Ed.), Algebraic Methodology and Software Technology, 8th Internat. Conf. AMAST 2000, Iowa City, Iowa, USA, May 20–27, 2000, Proceedings, Lecture Notes in Computer science, Vol. 1816, Springer, Berlin, 2000, pp. 407–421.
- [33] R. Bruni, A logic for modular descriptions of asynchronous and synchronized concurrent systems, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 225–236, <http://www.elsevier.nl/locate/entcs/volume15.html>.

- [34] R. Bruni, Tile logic for synchronized rewriting of concurrent systems, Ph.D. Thesis, Dipartimento di Informatica, Università di Pisa, 1999, Technical Report TD-1/99, <http://www.di.unipi.it/phd/tesi/tesi.1999/TD-1-99.ps.gz>.
- [35] R. Bruni, U. Montanari, Cartesian closed double categories, their lambda-notation, and the pi-calculus, in: Proc., 14th Annual IEEE Symp. on Logic in Computer Science, Trento, July 2–5, 1999. IEEE Computer Society, Silver Spring, MD, pp. 246–265.
- [36] R. Bruni, D. de Frutos-Escrig, N. Martí-Oliet, U. Montanari, Bisimilarity congruences for open terms and term graphs via tile logic, in: C. Palamidessi (Ed.), CONCUR 2000, Concurrency Theory, 11th Internat. Conf. University Park, PA, USA, August 22–25, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1877, Springer, Berlin, 2000, pp. 259–274.
- [37] R. Bruni, D. de Frutos-Escrig, N. Martí-Oliet, U. Montanari, Tile bisimilarity congruences for open terms and term graphs, Technical Report TR-00-06, Dipartimento di Informatica, Università di Pisa, 2000, <ftp://ftp.di.unipi.it/pub/techreports/TR-00-06.ps.Z>.
- [38] R. Bruni, J. Meseguer, U. Montanari, Implementing tile systems: some examples from process calculi, in: P. Degano, U. Vaccaro, G. Pirillo (Eds.), Proc. 6th Italian Conf. on Theoretical Computer Science, ICTCS'98, World Scientific, Singapore, 1998, pp. 168–179.
- [39] R. Bruni, J. Meseguer, U. Montanari, Internal strategies in a rewriting implementation of tile systems, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 95–116, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [40] R. Bruni, J. Meseguer, U. Montanari, Process and term tile logic. Technical Report SRI-CSL-98-06, Computer Science Laboratory, SRI International, July, 1998. Also TR-98-09, Dipartimento di Informatica, Università di Pisa, <ftp://ftp.di.unipi.it/pub/techreports/TR-98-09.ps.Z>.
- [41] R. Bruni, J. Meseguer, U. Montanari, Executable tile specifications for process calculi, in: J.-P. Finance (Ed.), Fundamental Approaches to Software Engineering, 2nd Internat. Conf. FASE'99, Held as Part of ETAPS'99, Amsterdam, The Netherlands, March 22–28, 1999, Proceedings, Lecture Notes in Computer Science, Vol. 1577, Springer, Berlin, 1999, pp. 60–76.
- [42] R. Bruni, J. Meseguer, U. Montanari, Symmetric monoidal and cartesian double categories as a semantic framework for tile logic, *Math. Struct. Comput. Sci.* (2002), to appear.
- [43] R. Bruni, U. Montanari, F. Rossi, An interactive semantics of logic programming, Manuscript, Dipartimento di Informatica, Università di Pisa, 2000, submitted for publication.
- [44] R. Bruni, U. Montanari, V. Sassone, Open ended systems, dynamic bisimulation and tile logic, in: J. van Leeuwen, O. Watanabe, M. Hagiya, P.D. Mosses, T. Ito (Eds.), Theoretical Computer Science: Exploring New Frontiers of Theoretical Informatics, Internat. Conf. IFIP TCS 2000 Sendai, Japan, August 17–19, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1872, Springer, Berlin, 2000, pp. 440–456.
- [45] G. Carabetta, P. Degano, F. Gadducci, CCS semantics via proved transition systems and rewriting logic, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 253–272, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [46] C. Castro, An approach to solving binary CSP using computational systems, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 245–264, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [47] C. Castro, Binary CSP solving as an inference process, in: Proc. of the 8th Internat. Conf. on Tools in Artificial Intelligence, ICTAI'96, Toulouse, France, November, 1996, pp. 462–463.
- [48] C. Castro, Constraint manipulation using rewrite rules and strategies, in: Proc. of the Second ESSLLI Student Session, 9th Europ. Summer School in Logic, Language and Information, ESSLLI'97, Aix-en-Provence, France, August, 1997, pp. 45–56.
- [49] C. Castro, Building constraint satisfaction problem solvers using rewrite rules and strategies, *Fund. Informat.* 34 (3) (1998) 263–293.

- [50] C. Castro, Une approche déductive de la résolution de problèmes de satisfaction de contraintes, Ph.D. Thesis, Université Henri Poincaré — Nancy I, 1998.
- [51] M.V. Cengarle, The rewriting logic institution, Technical Report 9801, Institut für Informatik, Ludwig-Maximilians-Universität München, May, 1998.
- [52] A. Ciampolini, E. Lamma, P. Mello, C. Stefanelli, Distributed logic objects: a fragment of rewriting logic and its implementation, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 109–124, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [53] H. Cirstea, Specifying authentication protocols in ELAN, in: Workshop on Modelling and Verification, Besançon, France, December, 1999.
- [54] H. Cirstea, Calcul de réécriture: fondements et applications, Ph.D. Thesis, Université Henri Poincaré — Nancy I, October, 2000.
- [55] H. Cirstea, C. Kirchner, Theorem proving using computational systems: the case of the B predicate prover, Presented at CCL'97 Workshop, Schloss Dagstuhl, Germany, September, 1997.
- [56] H. Cirstea, C. Kirchner, Using rewriting and strategies for describing the B predicate prover, in: B. Gramlich, F. Pfenning (Eds.), Proc. of the CADE-15 Workshop on Strategies in Automated Deduction, Lindau, Germany, July, 1998.
- [57] H. Cirstea, C. Kirchner, Combining higher-order and first-order computations using  $\rho$ -calculus: towards a semantics of ELAN, in: D. Gabbay, M. de Rijke (Eds.), Frontiers of Combining Systems 2, Research Studies Press/Wiley, New York, 1999, pp. 95–121.
- [58] H. Cirstea, C. Kirchner, The simply typed rewriting calculus, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 23–41, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [59] H. Cirstea, C. Kirchner, The rewriting calculus — part I, Logic J. Interest Group Pure Appl. Logics 9 (3) (2001) 363–399.
- [60] H. Cirstea, C. Kirchner, The rewriting calculus — part II, Logic J. Interest Group Pure Appl. Logics 9 (3) (2001) 401–434.
- [61] H. Cirstea, C. Kirchner, L. Liquori, Matching power, in: A. Middeldorp (Ed.), Rewriting Techniques and Applications, 12th International Conference, RTA 2001, Utrecht, The Netherlands, May 22–24, 2001, Proceedings, Lecture Notes in Computer Science, Vol. 2051, Springer, Berlin, 2001, pp. 77–92.
- [62] H. Cirstea, C. Kirchner, L. Liquori, The rho cube, in: F. Honsell, M. Miculan (Eds.), Foundations of Software Science and Computation Structures, 4th Internat. Conf. FOSSACS 2001, Held as Part of ETAPS 2001, Genova, Italy, April, 2001, Proceedings, Lecture Notes in Computer Science, Vol. 2030, Springer, Berlin, 2001, pp. 168–183.
- [63] M. Clavel, Reflection in general logics and in rewriting logic, with applications to the Maude language, Ph.D. Thesis, Universidad de Navarra, Spain, February, 1998.
- [64] M. Clavel, Reflection in general logics, rewriting logic, and Maude, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 317–328, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [65] M. Clavel, Reflection in Rewriting Logic: Metalogical Foundations and Metaprogramming Applications, CSLI Publications, Stanford, CA, 2000.
- [66] M. Clavel, J. Meseguer, Axiomatizing reflective logics and languages, in: G. Kiczales (Ed.), Proc. of Reflection'96, San Francisco, CA, April, 1996, pp. 263–288.
- [67] M. Clavel, J. Meseguer, Reflection and strategies in rewriting logic, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 125–147, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [68] M. Clavel, J. Meseguer, Internal strategies in a reflective logic, in: B. Gramlich, H. Kirchner (Eds.), Proc. of the CADE-14 Workshop on Strategies in Automated Deduction, Townsville, Australia, July 1997, pp. 1–12.

- [69] M. Clavel, J. Meseguer, Reflection in rewriting logic and its applications in the Maude language, in: Proc. IMSA'97, Information-Technology Promotion Agency, Japan, 1997, pp. 128–139.
- [70] M. Clavel, J. Meseguer, Reflection in conditional rewriting logic, Theoret. Comput. Sci. (2002), submitted for publication.
- [71] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, Metalevel computation in Maude, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 3–24, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [72] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, Language prototyping in the Maude metalanguage, Manuscript, Computer Science Laboratory, SRI International, November, 1998.
- [73] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, Maude as a metalanguage, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 237–250, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [74] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, Maude: specification and programming in rewriting logic, Manual distributed as documentation of the Maude system, Computer Science Laboratory, SRI International, January, 1999, <http://maude.csl.sri.com/manual>.
- [75] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, The Maude system, in: P. Narendran, M. Rusinowitch (Eds.), Rewriting Techniques and Applications, 10th Internat. Conf. RTA'99, Trento, Italy, July 2–4, 1999, Proceedings, Lecture Notes in Computer Science, Vol. 1631, Springer, Berlin, 1999, pp. 240–243.
- [76] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, A Maude tutorial, Tutorial distributed as documentation of the Maude system, Computer Science Laboratory, SRI International. Presented at the Europ. Joint Conf. on Theory and Practice of Software, ETAPS 2000, Berlin, Germany, March 25, 2000, <http://maude.csl.sri.com/tutorial>.
- [77] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, Towards Maude 2.0, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 297–318, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [78] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, Using Maude, in: T. Maibaum (Ed.), Fundamental Approaches to Software Engineering, 3rd Internat. Conf. FASE 2000, Held as Part of ETAPS 2000, Berlin, Germany, March/April 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1783, Springer, Berlin, 2000, pp. 371–374.
- [79] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, J.F. Quesada, Maude: specification and programming in rewriting logic, Theoret. Comput. Sci. 2002, submitted for publication.
- [80] M. Clavel, F. Durán, S. Eker, J. Meseguer, Building equational proving tools by reflection in rewriting logic, in: K. Futatsugi (Ed.), Proc. of the CafeOBJ Symposium '98, Numazu, Japan, CafeOBJ Project, April, 1998, <http://maude.csl.sri.com/papers>.
- [81] M. Clavel, F. Durán, S. Eker, J. Meseguer, Design and implementation of the Cafe prover and the Church–Rosser checker tools, Technical report, Computer Science Laboratory, SRI International, March, 1998, <http://maude.csl.sri.com/papers>.
- [82] M. Clavel, F. Durán, S. Eker, J. Meseguer, Building equational proving tools by reflection in rewriting logic, in: K. Futatsugi, A.T. Nakagawa, T. Tamai (Eds.), Cafe: an Industrial-Strength Algebraic Formal Method, Elsevier, Amsterdam, 2000, pp. 1–31, <http://maude.csl.sri.com/papers>.
- [83] M. Clavel, F. Durán, S. Eker, J. Meseguer, M.-O. Stehr, Maude as a formal meta-tool, in: J.M. Wing, J. Woodcock, J. Davies (Eds.), FM'99 — Formal Methods, World Congress on Formal Methods in the Development of Computing Systems, Toulouse, France, September 20–24, 1999, Proceedings, Vol. II, Lecture Notes in Computer Science, Vol. 1709, Springer, Berlin, 1999, pp. 1684–1703.

- [84] M. Clavel, F. Durán, N. Martí-Oliet, Polytypic programming in Maude, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 339–360, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [85] M. Clavel, S. Eker, P. Lincoln, J. Meseguer, Principles of Maude, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 65–89, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [86] A. Corradini, F. Gadducci, CPO models for infinite term rewriting, in: V.S. Alagar, M. Nivat (Eds.), Algebraic Methodology and Software Technology, 4th Internat. Conf. AMAST'95, Montreal, Canada, July 3–7, 1995, Proceedings, Lecture Notes in Computer Science, Vol. 936, Springer, Berlin, 1995, pp. 368–384.
- [87] A. Corradini, F. Gadducci, U. Montanari, Relating two categorical models of term rewriting, in: J. Hsiang (Ed.), Rewriting Techniques and Applications, 6th Internat. Conf. RTA'95, Kaiserslautern, Germany, April 5–7, 1995, Proceedings, Lecture Notes in Computer Science, Vol. 914, Springer, Berlin, 1995, pp. 225–240.
- [88] A. Corradini, R. Heckel, U. Montanari, Tile transition systems as structured coalgebras, in: G. Ciobanu, G. Paun (Eds.), Fundamentals of Computation Theory, 12th Internat. Symp. FCT'99, Iasi, Romania, August 30–September 3, 1999, Proceedings, Lecture Notes in Computer Science, Vol. 1684, Springer, Berlin, 1999, pp. 13–38.
- [89] P. Degano, F. Gadducci, C. Priami, A causal semantics for CCS via rewriting logic, Manuscript, Dipartimento di Informatica, Università di Pisa, 2000, submitted for publication.
- [90] G. Denker, From rewrite theories to temporal logic theories, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 273–294, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [91] G. Denker, M. Gogolla, Translating TROLL light concepts to Maude, in: H. Ehrig, F. Orejas (Eds.), Recent Trends in Data Type Specification, 9th Workshop on Specification of Abstract Data Types Joint with the 4th COMPASS Workshop, Caldes de Malavella, Spain, October 1992, Selected Papers, Lecture Notes in Computer Science, Vol. 785, Springer, Berlin, 1994, pp. 173–187.
- [92] G. Denker, J. Millen, CAPSL and CIL language design: a common authentication protocol specification language and its intermediate language, Technical Report SRI-CSL-99-02, Computer Science Laboratory, SRI International, 1999, [http://www.csl.sri.com/~denker/pub\\_99.html](http://www.csl.sri.com/~denker/pub_99.html).
- [93] G. Denker, J. Millen, CAPSL intermediate language, in: N. Heintze, E. Clarke (Eds.), Proc. of Workshop on Formal Methods and Security Protocols, FMSP'99, July 1999, Trento, Italy, 1999, <http://www.cs.bell-labs.com/who/nch/fmsp99/program.html>.
- [94] G. Denker, J. Millen, CAPSL integrated protocol environment, in: D. Maughan, G. Koob, S. Saydjari (Eds.), Proc. DARPA Information Survivability Conference and Exposition, DISCEX 2000, Hilton Head Island, SC, January 25–27, 2000, IEEE Computer Society Press, 2000, Silver Spring, MD, pp. 207–222, <http://schafercorp-ballston.com/discex/>.
- [95] G. Denker, J. Millen, The CAPSL integrated protocol environment, Technical Report SRI-CSL-2000-02, Computer Science Laboratory, SRI International, 2000, [http://www.csl.sri.com/~denker/pub\\_99.html](http://www.csl.sri.com/~denker/pub_99.html).
- [96] G. Denker, J.J. García-Luna-Aceves, J. Meseguer, P.C. Ölveczky, J. Raju, B. Smith, C.L. Talcott, Specification and analysis of a reliable broadcasting protocol in Maude, in: B. Hajek, R.S. Sreenivas (Eds.), Proc. 37th Allerton Conference on Communication, Control and Computation, University of Illinois, 1999, pp. 738–747, <http://maude.csl.sri.com/casestudies/rbp>.
- [97] G. Denker, J.J. García-Luna-Aceves, J. Meseguer, P.C. Ölveczky, J. Raju, B. Smith, C.L. Talcott, Specifying a reliable broadcasting protocol in Maude, Technical report, Computer Science Laboratory, SRI International, 1999, <http://www.csl.sri.com/casestudies/rbp>.
- [98] G. Denker, J. Meseguer, C.L. Talcott, Protocol specification and analysis in Maude, in: N. Heintze, J. Wing (Eds.), Proc. of Workshop on Formal Methods and Security Protocols, June 25, 1998, Indianapolis, Indiana, 1998, <http://www.cs.bell-labs.com/who/nch/fmsp/index.html>.

- [99] G. Denker, J. Meseguer, C.L. Talcott, Formal specification and analysis of active networks and communication protocols: the Maude experience, in: D. Maughan, G. Koob, S. Saydjari (Eds.), Proc. DARPA Information Survivability Conference and Exposition, DISCEX 2000, Hilton Head Island, SC, January 25–27, 2000, IEEE Computer Society Press, Silver Spring, MD, 2000, pp. 251–265, <http://schafercorp-ballston.com/discex/>.
- [100] G. Denker, J. Meseguer, C.L. Talcott, Rewriting semantics of meta-objects and composable distributed services, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 407–427, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [101] E. Deplagne, Sequent calculus viewed modulo, in: C. Pilière (Ed.), Proc. of the 5th ESSLLI Student Session, University of Birmingham, August, 2000, pp. 66–76, <http://www.loria.fr/publications/2000/A00-R-256/A00-R-256.ps>.
- [102] R. Diaconescu, Behavioural rewriting logic: semantic foundations and proof theory, Manuscript, October, 1996, submitted for publication, <http://ldl-www.jaist.ac.jp/cafeobj/documents.html>.
- [103] R. Diaconescu, Foundations of behavioural specification in rewriting logic, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 225–244, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [104] R. Diaconescu, K. Futatsugi, CafeOBJ Report, The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification, AMAST Series in Computing, Vol. 6, World Scientific, Singapore, 1998.
- [105] R. Diaconescu, K. Futatsugi, Logical foundations of CafeOBJ, Theoret. Comput. Sci. (2002), submitted for publication.
- [106] R. Diaconescu, K. Futatsugi, S. Iida, Component-based algebraic specification and verification in CafeOBJ, in: J.M. Wing, J. Woodcock, J. Davies (Eds.), FM'99 — Formal Methods, World Congress on Formal Methods in the Development of Computing Systems, Toulouse, France, September 20–24, 1999 Proceedings, Vol. II, Lecture Notes in Computer Science, Vol. 1709, Springer, Berlin, 1999, pp. 1644–1663.
- [107] R. Diaconescu, K. Futatsugi, S. Iida, CafeOBJ jewels, in: K. Futatsugi, A.T. Nakagawa, T. Tamai (Eds.), Cafe: an Industrial-Strength Algebraic Formal Method, Elsevier, Amsterdam, 2000, pp. 33–60.
- [108] R. Diaconescu, K. Futatsugi, M. Ishisone, T. Sawada, A.T. Nakagawa, An overview of CafeOBJ, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 75–88, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [109] G. Dowek, T. Hardin, C. Kirchner, Theorem proving modulo, Technical Report RR-3400, INRIA, Institut National de Recherche en Informatique et en Automatique, April 1998, <ftp://ftp.inria.fr/INRIA/publication/RR/RR-3400.ps.gz>.
- [110] G. Dowek, T. Hardin, C. Kirchner, Higher order unification via explicit substitutions, Inform. Comput. 157 (1/2) (2000) 183–235.
- [111] G. Dowek, T. Hardin, C. Kirchner, HOL- $\lambda\sigma$ : an intentional first-order expression of higher-order logic, Math. Struct. Comput. Sci. 11 (1) (2001) 21–45.
- [112] H. Dubois, H. Kirchner, Actions and plans in ELAN, in: B. Gramlich, F. Pfenning (Eds.), Proc. of the CADE-15 Workshop on Strategies in Automated Deduction, Lindau, Germany, July 1998, pp. 35–45, also Technical Report LORIA 98-R-275.
- [113] H. Dubois, H. Kirchner, Modelling planning problems with rules and strategies, Technical Report 99-R-029, LORIA, Nancy, France, March, 1999, Poster Session at JFPLC'99, Lyon, France, June, 1999.
- [114] H. Dubois, H. Kirchner, Objects, rules, and strategies in ELAN, in: Proc. of the 2nd AMAST Workshop on Algebraic Methods in Language Processing, Iowa City, Iowa, USA, 1999.
- [115] H. Dubois, H. Kirchner, Rule based programming with constraints and strategies, in: K.R. Apt, A. Kakas, E. Monfroy, F. Rossi (Eds.), New Trends in Constraints, Joint ERCIM/Compulog Net Workshop, Paphos, Cyprus, October 25–27, 1999, Selected Papers, Lecture Notes in Artificial Intelligence, Vol. 1865, Springer, Berlin, 2000, pp. 274–297.



- [116] F. Durán, A reflective module algebra with applications to the Maude language, Ph.D. Thesis, Universidad de Málaga, Spain, June, 1999, <http://maude.csl.sri.com/papers>.
- [117] F. Durán, Coherence checker and completion tools for Maude specifications, Manuscript, Computer Science Laboratory, SRI International, 2000, <http://maude.csl.sri.com/papers>.
- [118] F. Durán, The extensibility of Maude's module algebra, in: T. Rus (Ed.), Algebraic Methodology and Software Technology, 8th Internat. Conf. AMAST 2000, Iowa City, Iowa, USA, May 20–27, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1816, Springer, Berlin, 2000, pp. 422–437.
- [119] F. Durán, Termination checker and Knuth–Bendix completion tools for Maude equational specifications, Manuscript, Computer Science Laboratory, SRI International, 2000, <http://maude.csl.sri.com/papers>.
- [120] F. Durán, J. Meseguer, An extensible module algebra for Maude, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 185–206, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [121] F. Durán, J. Meseguer, The Maude specification of Full Maude, Manuscript, Computer Science Laboratory, SRI International, February, 1999, <http://maude.csl.sri.com/papers>.
- [122] F. Durán, J. Meseguer, Structured theories and institutions. in: M. Hofmann, G. Rosolini, D. Pavlović (Eds.), Proc. of 8th Conference on Category Theory and Computer Science, Edinburgh, Scotland, September 1999, Electronic Notes in Theoretical Computer Science, Vol. 29, Elsevier, Amsterdam, 1999, pp. 71–90, <http://www.elsevier.nl/locate/entcs/volume29.html>.
- [123] F. Durán, J. Meseguer, A Church–Rosser checker tool for Maude equational specifications, Manuscript, Computer Science Laboratory, SRI International, 2000, <http://maude.csl.sri.com/papers>.
- [124] F. Durán, J. Meseguer, Parameterized theories and views in Full Maude 2.0, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 319–337, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [125] F. Durán, J. Meseguer, Structured theories and institutions, Manuscript, Computer Science Laboratory, SRI International, 2000, submitted for publication.
- [126] F. Durán, J. Meseguer, Maude's module algebra, Theoret. Comput. Sci. (2002), to appear, <http://maude.csl.sri.com/papers>.
- [127] F. Durán, A. Vallecillo, Writing ODP enterprise specifications in Maude, in: Proc. Workshop on Open Distributed Processing: Enterprise, Computation, Knowledge, Engineering and Realisation, WOODPECKER 2001, Setúbal, Portugal, July, 2001, <http://www.lcc.uma.es/~av/Publicaciones/01/ITI-2001-8.pdf>.
- [128] F. Durán, S. Eker, P. Lincoln, J. Meseguer, Principles of mobile Maude, in: D. Kotz, F. Mattern (Eds.), Agent Systems, Mobile Agents, and Applications, 2nd Internat. Symp. on Agent Systems and Applications and 4th Internat. Symp. on Mobile Agents, ASA/MA 2000, Zurich, Switzerland, September 13–15, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1882, Springer, Berlin, 2000, pp. 73–85.
- [129] S. Eker, Fast matching in combination of regular equational theories, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 90–108, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [130] S. Eker, Term rewriting with operator evaluation strategy. in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 45–62, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [131] J.L. Fernández, A. Tóval, Can intuition become rigorous? Foundations for UML model verification tools, in: F.M. Titsworth (Ed.), Internat. Symp. on Software Reliability Engineering, San Jose, California, October 8–11, IEEE Press, New York, 2000, pp. 344–355.
- [132] J.L. Fernández, A. Tóval, Seamless formalizing the UML semantics through metamodels, in: K. Siau, T. Halpin (Eds.), Unified Modeling Language: Systems Analysis, Design, and Development Issues, Idea Group Publishing, 2001, pp. 224–248.



- [133] G. Ferrari, U. Montanari, A tile-based coordination view of asynchronous pi-calculus, in: I. Privara, P. Ruzicka (Eds.), *Mathematical Foundations of Computer Science 1997*, 22nd Internat. Symp. MFCS'97, Bratislava, Slovakia, August 25–29, 1997, Proceedings, Lecture Notes in Computer Science, Vol. 1295, Springer, Berlin, 1997, pp. 52–70.
- [134] G. Ferrari, U. Montanari, Tiles for concurrent and located calculi, in: C. Palamidessi, J. Parrow (Eds.), *Proc. EXPRESS'97*, Electronic Notes in Theoretical Computer Science, Vol. 7, Elsevier, Amsterdam, 1997. <http://www.elsevier.nl/locate/entcs/volume7.html>.
- [135] G. Ferrari, U. Montanari, Tile formats for located and mobile systems, *Inform. Comput.* 156 (1/2) (2000) 173–235.
- [136] J.L. Fiadeiro, T. Maibaum, N. Martí-Oliet, J. Meseguer, I. Pita, Towards a verification logic for rewriting logic, in: D. Bert, C. Choppy, P. Mosses (Eds.), *Recent Trends in Algebraic Development Techniques*, 14th Internat. Workshop, WADT'99, Chateau de Bonas, France, September 15–18, 1999, Selected Papers, Lecture Notes in Computer Science, Vol. 1827, Springer, Berlin, 2000, pp. 438–458.
- [137] B. Fischer, G. Roşu, Interpreting abstract interpretations in membership equational logic, Technical Report RIACS 01.16, Research Institute for Advanced Computer Science, May, 2001.
- [138] K. Futatsugi (Ed.), *Proc. of the CafeOBJ Symposium '98*, Numazu, Japan, CafeOBJ Project, April, 1998.
- [139] K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications*, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000. <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [140] K. Futatsugi, A.T. Nakagawa, An overview of CAFE specification environment — an algebraic approach for creating, verifying, and maintaining formal specifications over networks, in: 1st Internat. Conf. on Formal Engineering Methods, Hiroshima, Japan, November, 1997, <http://ldl-www.jaist.ac.jp/cafeobj/documents.html>.
- [141] K. Futatsugi, T. Sawada, Cafe as an extensible specification environment, in: *Proc. of the Kunming International CASE Symposium*, Kunming, China, November, 1994.
- [142] K. Futatsugi, J.A. Goguen, J. Meseguer (Eds.), *OBJ/CafeOBJ/Maude Workshop at Formal Methods '99: Formal Specification, Proof, and Applications*, Theta, Bucharest, 1999.
- [143] K. Futatsugi, A.T. Nakagawa, T. Tamai (Eds.), *Cafe: an Industrial-Strength Algebraic Formal Method*, Elsevier, Amsterdam, 2000.
- [144] F. Gadducci, On the algebraic approach to concurrent term rewriting, Ph.D. Thesis, Dipartimento di Informatica, Università di Pisa, March, 1996, Technical Report TD-2/96.
- [145] F. Gadducci, U. Montanari, Enriched categories as models of computation, in: A. De Santis (Ed.), *Proc. 5th Italian Conf. on Theoretical Computer Science*, Ravello, World Scientific, Singapore, 1995, pp. 20–42.
- [146] F. Gadducci, U. Montanari, Tiles, rewriting rules, and CCS, in: J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Applications*, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 1–19, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [147] F. Gadducci, U. Montanari, The tile model, in: G. Plotkin, C. Stirling, M. Tofte (Eds.), *Proof, Language and Interaction: Essays in Honour of Robin Milner*, The MIT Press, Cambridge, MA, 2000, <http://www.di.unipi.it/~ugo/festschrift.ps>.
- [148] F. Gadducci, U. Montanari, Comparing logics for rewriting: rewriting logic, action calculi and tile logic, *Theoret. Comput. Sci.* (2002), submitted for publication.
- [149] T. Genet, Contraintes d'ordre et automates d'arbres pour les preuves de terminaison, Ph.D. Thesis, Université Henri Poincaré — Nancy I, 1998.
- [150] T. Genet, Decidable approximations of sets of descendants and sets of normal forms, in: T. Nipkow (Ed.), *Rewriting Techniques and Applications*, 9th International Conference, RTA'98, Tsukuba, Japan, March 30–April 1, 1998, Proceedings, Lecture Notes in Computer Science, Vol. 1379, Springer, Berlin, 1998, pp. 151–165.
- [151] F. Giunchiglia, The OMRS project: state of the art and future developments, in: C. Kirchner, H. Kirchner (Eds.), *Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications*, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, <http://www.elsevier.nl/locate/entcs/volume15.html>.

- [152] J.A. Goguen, K. Lin, G. Roşu, Behavioral and coinductive rewriting, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 1–22, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [153] J.C. González-Moreno, M.T. Hortalá-González, F.J. López-Fraguas, M. Rodríguez-Artalejo, An approach to declarative programming based on a rewriting logic, *J. Logic Programming* 40 (1999) 47–87.
- [154] N.A. Harman, Correctness and verification of hardware systems using Maude, Technical Report 3-2000, Department of Computer Science, University of Wales Swansea, 2000, <http://www.compsci.swan.ac.uk/reports/yr2000/CSR3-2000.pdf>.
- [155] N.A. Harman, Verifying a simple pipelined microprocessor using Maude, Technical Report 4-2000, Department of Computer Science, University of Wales Swansea, 2000, <http://www.compsci.swan.ac.uk/reports/yr2000/CSR4-2000.pdf>.
- [156] N.A. Harman, Verifying a microprocessor using Maude, in: M. Cerioli, P.D. Mosses, G. Reggio (Eds.), Proc. WADT/CoFI'01, 15th Internat. Workshop on Algebraic Development Techniques and General Workshop of the CoFI WG, Genova, Italy, April 1–3, 2001.
- [157] K. Havelund, G. Roşu, Java PathExplorer — a runtime verification tool, in: Proc. 6th Internat. Symp. on Artificial Intelligence, Robotics and Automation in Space, ISAIRAS'01, Montreal, Canada, June, 18–22, 2001.
- [158] K. Havelund, G. Roşu, Monitoring Java programs with Java PathExplorer, in: Proc. 1st Workshop on Runtime Verification, RV'01, Paris, France, July 23, 2001, Electronic Notes in Theoretical Computer Science, Vol. 55 (2), Elsevier, Amsterdam, 2001, <http://www.elsevier.nl/locate/entcs/volume55.html>.
- [159] K. Havelund, G. Roşu, Monitoring programs using rewriting, Technical report, Research Institute for Advanced Computer Science, 2001.
- [160] K. Havelund, G. Roşu, Testing linear temporal logic formulae on finite execution traces, Technical Report RIACS 01.08, Research Institute for Advanced Computer Science, May, 2001.
- [161] H. Hilberdink, Foundations for rewriting logic, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 43–69, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [162] S. Iida, M. Matsumoto, R. Diaconescu, K. Futatsugi, D. Lucanu, Concurrent object composition in CafeOBJ, Technical Report JAIST IS-RR-98-0009S, Japan Advanced Institute of Science and Technology, 1998, <http://ldl-www.jaist.ac.jp/cafeobj/documents.html>.
- [163] H. Ishikawa, K. Futatsugi, T. Watanabe, An operational semantics of GAEA in CafeOBJ, in: K. Futatsugi, J.A. Goguen, J. Meseguer (Eds.), OBJ/CafeOBJ/Maude Workshop at Formal Methods '99: Formal Specification, Proof, and Applications, Theta, Bucharest, 1999, pp. 213–227.
- [164] H. Ishikawa, J. Meseguer, T. Watanabe, K. Futatsugi, H. Nakashima, On the semantics of GAEA — an object-oriented specification of a concurrent reflective language in rewriting logic, in: Proc. IMSA'97, Information-Technology Promotion Agency, Japan, 1997, pp. 70–109.
- [165] M. Ishisone, T. Sawada, Brute: brute force rewriting engine, in: K. Futatsugi (Ed.), Proc. of the CafeOBJ Symposium '98, Numazu, Japan, CafeOBJ Project, April, 1998.
- [166] J.-P. Jouannaud, Membership equational logic, calculus of inductive constructions, and rewrite logic, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 89–94, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [167] H. Kirchner, On the use of constraints in automated deduction, in: A. Podelski (Ed.), Constraint Programming: Basics and Trends, 1994 Chatillon Spring School, Chatillon-sur-Seine, France, May 16–20, 1994, Selected Papers, Lecture Notes in Computer Science, Vol. 910, Springer, Berlin, 1995, pp. 128–146.
- [168] H. Kirchner, Some extensions of rewriting, in: H. Comon, J.-P. Jouannaud (Eds.), Term Rewriting, French Spring School of Theoretical Computer Science, Font Romeaux, France, May 17–21, 1993, Advanced Course, Lecture Notes in Computer Science, Vol. 909, Springer, Berlin, 1995, pp. 54–73.

- [169] H. Kirchner, ELAN, in: F. Fages (Ed.) JFPLC'99, Journées Francophones de Programmation Logique et Programmation par Contraintes, Lyon, France, HERMES Science Publications, May, 1999, pp. 241–248, Also Technical Report 99-R-129, LORIA, Nancy, France.
- [170] H. Kirchner, Term rewriting, in: E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner (Eds.), Algebraic Foundations of Systems Specification, IFIP State-of-the-Art Reports, Springer, Berlin, 1999, pp. 273–320.
- [171] H. Kirchner, I. Gnaedig, Termination and normalisation under strategy — proofs in ELAN, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 93–115, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [172] C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [173] H. Kirchner, P.-E. Moreau, Prototyping completion with constraints using computational systems, in: J. Hsiang (Ed.), Rewriting Techniques and Applications, 6th Internat. Conf. RTA'95, Kaiserslautern, Germany, April 5–7, 1995, Proceedings, Lecture Notes in Computer Science, Vol. 914, Springer, Berlin, 1995, pp. 438–443.
- [174] H. Kirchner, P.-E. Moreau, A reflective extension of ELAN, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 148–167, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [175] H. Kirchner, P.-E. Moreau, Non-deterministic computations in ELAN, in: J.L. Fiadeiro (Ed.), Recent Trends in Algebraic Development Techniques, 13th Internat. Workshop, WADT'98, Lisbon, Portugal, April 2–4, 1998, Selected Papers, Lecture Notes in Computer Science, Vol. 1589, Springer, Berlin, 1999, pp. 168–182.
- [176] H. Kirchner, P.-E. Moreau, Promoting rewriting to a programming language: a compiler for non-deterministic rewrite programs in associative-commutative theories, *J. Funct. Programming* 11 (2) (2001) 207–251.
- [177] H. Kirchner, C. Ringeissen, Combining symbolic constraint solvers on algebraic domains, *J. Symbolic Comput.* 18 (2) (1994) 113–155.
- [178] H. Kirchner, C. Ringeissen, Constraint solving by narrowing in combined algebraic domains, in: P. van Hentenryck (Ed.), Proc. 11th Internat. Conf. on Logic Programming, Santa Margherita Ligure, Italy, The MIT Press, Cambridge, MA, 1994, pp. 617–631.
- [179] H. Kirchner, C. Ringeissen, Executing CASL equational specifications with the ELAN rewrite engine, Technical Report 99-R-278, LORIA, Nancy, France, 1999.
- [180] C. Kirchner, C. Ringeissen, Rule-based constraint programming, *Fund. Inform.* 34 (3) (1998) 225–262.
- [181] H. Kirchner, L. Vigneron, Deduction with constraints for theory reasoning: completeness and simplification problems, in: Proc. CADE-12 Workshop: Theory Reasoning in Automated Deduction, Nancy, France, 1994.
- [182] C. Kirchner, H. Kirchner, M. Vittek, Implementing computational systems with constraints, in: P. Kanellakis, J.-L. Lassez, V. Saraswat (Eds.), Proc. 1st Workshop on Principles and Practice of Constraint Programming, Brown University, Providence, RI, USA, 1993, pp. 166–175.
- [183] C. Kirchner, H. Kirchner, M. Vittek, Designing constraint logic programming languages using computational systems, in: V. Saraswat, P. van Hentenryck (Eds.), Principles and Practice of Constraint Programming: The Newport Papers, The MIT Press, Cambridge, MA, 1995, pp. 133–160.
- [184] A. Knapp, Case studies with CafeOBJ, in: K. Futatsugi (Ed.), Proc. of the CafeOBJ Symposium'98, Numazu, Japan, CafeOBJ Project, April, 1998.
- [185] A. Knapp, Generating rewrite theories from UML collaborations, in: K. Futatsugi, A.T. Nakagawa, T. Tamai (Eds.), Cafe: an Industrial-Strength Algebraic Formal Method, Elsevier, Amsterdam, 2000, pp. 97–120.
- [186] A. Knapp, A Formal Approach to Object-Oriented Software Engineering, Shaker Verlag, Aachen, Germany, 2001, Ph.D. Thesis, Institut für Informatik, Universität München, 2000.

- [187] P. Kosiuczenko, M. Wirsing, Timed rewriting logic for the specification of time-sensitive systems, in: H. Schwichtenberg (Ed.), *Logic of Computation, Proc. of the NATO Advanced Study Institute on Logic of Computation, Held in Marktoberdorf, Germany, July 25–August 6, 1997*, NATO ASI Series F: Computer and Systems Sciences, Vol. 157, Springer, Berlin, 1997, pp. 229–264.
- [188] P. Kosiuczenko, M. Wirsing, Timed rewriting logic with an application to object-based specification, *Sci. Comput. Programming* 28 (2–3) (1997) 225–246.
- [189] M. Kulaš, C. Beirle, Defining standard prolog in rewriting logic, in: K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000*, *Electronic Notes in Theoretical Computer Science*, Vol. 36, Elsevier, Amsterdam, 2000, pp. 155–171, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [190] H. Kuruma, K. Futatsugi, Incremental specification based on the combination of data types in: K. Futatsugi, J.A. Goguen, J. Meseguer (Eds.), *OBJ/CafeOBJ/Maude Workshop at Formal Methods'99: Formal Specification, Proof, and Applications, Theta, 1999, Bucharest*, pp. 95–114.
- [191] C. Landauer, Discrete event systems in rewriting logic, in: J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA' 96, Asilomar, CA, September 3–6, 1996*, *Electronic Notes in Theoretical Computer Science*, Vol. 4, Elsevier, Amsterdam, 1996, pp. 309–320, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [192] C. Laneve, U. Montanari, Axiomatizing permutation equivalence in the  $\lambda$ -calculus, in: H. Kirchner, G. Levi (Eds.), *Algebraic and Logic Programming, 3rd Internat. Conf., Volterra, Italy, September 2–4, 1992*, *Proceedings, Lecture Notes in Computer Science*, Vol. 632, Springer, Berlin, 1992, pp. 350–363.
- [193] C. Laneve, U. Montanari, Axiomatizing permutation equivalence, *Math. Struct. Comput. Sci.* 6 (3) (1996) 219–249.
- [194] U. Lechner, Object-oriented specifications of distributed systems in the  $\mu$ -calculus and Maude, in: J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA' 96, Asilomar, CA, September 3–6, 1996*, *Electronic Notes in Theoretical Computer Science*, Vol. 4, Elsevier, Amsterdam, 1996, pp. 384–403, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [195] U. Lechner, Object-oriented specification of distributed systems. Ph.D. Thesis, Fakultät für Mathematik und Informatik, Universität Passau, June, 1997.
- [196] U. Lechner, Constructs, concepts, and criteria for reuse in concurrent object-oriented languages, in: E. Astesiano (Ed.), *Fundamental Approaches to Software Engineering, 1st Internat. Conf., FASE'98, Held as Part of ETAPS'98, Lisbon, Portugal, March 28–April 4, 1998*, *Proceedings, Lecture Notes in Computer Science*, Vol. 15, Springer, Berlin, 1998, pp. 171–187.
- [197] U. Lechner, Object-oriented specification of distributed systems, in: C. Kirchner, H. Kirchner (Eds.), *Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998*, *Electronic Notes in Theoretical Computer Science*, Elsevier, Amsterdam, 1998, pp. 405–414, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [198] U. Lechner, C. Lengauer, Modal  $\mu$ -Maude — properties and specification of concurrent objects, in: B. Freitag, C.B. Jones, C. Lengauer, H.-J. Schek (Eds.), *Object Orientation with Parallelism and Persistence*, Kluwer Academic Publishers, Dordrecht, 1996, pp. 41–62.
- [199] U. Lechner, C. Lengauer, F. Nickl, M. Wirsing, (Objects + concurrency) & reusability — a proposal to circumvent the inheritance anomaly, in: P. Cointe (Ed.), *ECOOP'96 — Object-Oriented Programming, 10th Europ. Conf. Linz, Austria, July 8–12, 1996*, *Proceedings, Lecture Notes in Computer Science*, Vol. 1098, Springer, Berlin, 1996, pp. 232–247.
- [200] U. Lechner, C. Lengauer, M. Wirsing, An object-oriented airport, in: E. Astesiano, G. Reggio, A. Tarlecki (Eds.), *Recent Trends in Data Type Specification, 10th Workshop on Specification of Abstract Data Types Joint with the 5th COMPASS Workshop, S. Margherita, Italy, May 30–June 3, 1994*, *Selected Papers, Lecture Notes in Computer Science*, Vol. 906, Springer, Berlin, 1995, pp. 351–367.
- [201] M. Leucker, T. Noll, Rewriting logic as a framework for generic verification tools, in: K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000*, *Electronic Notes in Theoretical Computer Science*, Vol. 36, Elsevier, Amsterdam, 2000, pp. 117–133, <http://www.elsevier.nl/locate/entcs/volume36.html>.

- [202] P. Lincoln, J. Meseguer, Strategic reflection, in: B. Gramlich, F. Pfenning (Eds.), *Proc. of the CADE-15 Workshop on Strategies in Automated Deduction*, Lindau, Germany, July, 1998, pp. 3–9.
- [203] P. Lincoln, N. Martí-Oliet, J. Meseguer, Specification, transformation, and programming of concurrent systems in rewriting logic, in: G.E. Blelloch, K.M. Chandy, S. Jagannathan (Eds.), *Specification of Parallel Algorithms*, DIMACS Workshop, May 9–11, 1994, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 18, American Mathematical Society, Providence, RI, 1994, pp. 309–339.
- [204] P. Lincoln, N. Martí-Oliet, J. Meseguer, L. Ricciulli, Compiling rewriting onto SIMD and MIMD/SIMD machines, in: C. Halatsis, D. Maritsas, G. Philokyprou, S. Theodoridis (Eds.), *PARLE'94 Parallel Architectures and Languages Europe*, 6th Internat. PARLE Conf., Athens, Greece, July 4–8, 1994, *Proceedings, Lecture Notes in Computer Science*, Vol. 817, Springer, Berlin, 1994, pp. 37–48.
- [205] M. Lowry, T. Pressburger, G. Roşu, Certifying domain-specific policies, Technical report, Research Institute for Advanced Computer Science, 2001.
- [206] D. Lucanu, Algebraic specification of object aggregation — an event oriented approach, in: K. Futatsugi, J.A. Goguen, J. Meseguer (Eds.), *OBJ/CafeOBJ/Maude Workshop at Formal Methods'99: Formal Specification, Proof, and Applications*, Theta, Bucharest, 1999, pp. 115–132.
- [207] D. Lucanu, Relaxed models for rewriting logic, Manuscript, July, 2000, submitted for publication.
- [208] N. Martí-Oliet, J. Meseguer, Rewriting logic as a logical and semantic framework, D. Gabbay (Ed.), *Handbook of Philosophical Logic*, Second Edition, Vol. 9, Kluwer Academic Publishers, Dordrecht, 2002, <http://maude.csl.sri.com/papers>.
- [209] N. Martí-Oliet, J. Meseguer, General logics and logical frameworks, in: D.M. Gabbay (Ed.), *What is a Logical System?*, *Studies in Logic and Computation*, Vol. 4, Oxford University Press, Oxford, 1994, pp. 355–392.
- [210] N. Martí-Oliet, J. Meseguer, Rewriting logic as a logical and semantic framework, in: J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Application*, WRLA'96, Asilomar, CA, September 3–6, 1996, *Electronic Notes in Theoretical Computer Science*, Vol. 4, Elsevier, Amsterdam, 1996, pp. 189–224, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [211] N. Martí-Oliet, J. Meseguer, Action and change in rewriting logic, in: R. Pareschi, B. Fronhöfer (Eds.), *Dynamic Worlds: From the Frame Problem to Knowledge Management*, *Applied Logic Series*, Vol. 12, Kluwer Academic Publishers, Dordrecht, 1999, pp. 1–53.
- [212] I.A. Mason, C.L. Talcott, A semantics preserving actor translation, in: P. Degano, R. Gorrieri, A. Marchetti-Spaccamela (Eds.), *Automata, Languages and Programming*, 24th Internat. Colloquium, ICALP'97, Bologna, Italy, July 1997, *Proceedings, Lecture Notes in Computer Science*, Vol. 1256, Springer, Berlin, 1997, pp. 369–378.
- [213] I.A. Mason, C.L. Talcott, Simple network protocol simulation within Maude, in: K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications*, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, *Electronic Notes in Theoretical Computer Science*, Vol. 36, Elsevier, Amsterdam, 2000, pp. 277–294, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [214] J. Meseguer, A logical theory of concurrent objects, in: N. Meyrowitz (Ed.), *Proc. ECOOP-OOPSLA'90 Conf. on Object-Oriented Programming*, Ottawa, Canada, October, 1990, ACM Press, New York, 1990, pp. 101–115.
- [215] J. Meseguer, Rewriting as a unified model of concurrency, Technical Report SRI-CSL-90-02R, SRI International, Computer Science Laboratory, February, 1990, Revised June, 1990. Appendices on functorial semantics have not been published elsewhere.
- [216] J. Meseguer, Rewriting as a unified model of concurrency, in: J.C.M. Baeten, J.W. Klop (Eds.), *CONCUR'90, Theories of Concurrency: Unification and Extension*, Amsterdam, The Netherlands, August 1990, *Proceedings, Lecture Notes in Computer Science*, Vol. 458, Springer, Berlin, 1990, pp. 384–400.
- [217] J. Meseguer, Conditional rewriting logic: deduction, models and concurrency, in: S. Kaplan, M. Okada (Eds.), *Conditional and Typed Rewriting Systems*, 2nd Internat. CTRS Workshop, Montreal, Canada, June 11–14, 1990, *Proceedings, Lecture Notes in Computer Science*, Vol. 516, Springer, Berlin, 1991, pp. 64–91.
- [218] J. Meseguer, Conditional rewriting logic as a unified model of concurrency, *Theoret. Comput. Sci.* 96 (1) (1992) 73–155.

- [219] J. Meseguer, Multiparadigm logic programming, in: H. Kirchner, G. Levi (Eds.), *Algebraic and Logic Programming*, 3rd Internat. Conf., Volterra, Italy, September 2–4, 1992, Proceedings, Lecture Notes in Computer Science, Vol. 632, Springer, Berlin, 1992, pp. 158–200.
- [220] J. Meseguer, A logical theory of concurrent objects and its realization in the Maude language, in: G. Agha, P. Wegner, A. Yonezawa (Eds.), *Research Directions in Concurrent Object-Oriented Programming*, The MIT Press, Cambridge, MA, 1993, pp. 314–390.
- [221] J. Meseguer, Solving the inheritance anomaly in concurrent object-oriented programming, in: O.M. Nierstrasz (Ed.), *ECOOP'93 — Object-Oriented Programming*, 7th Europ. Conf. Kaiserslautern, Germany, July 26–30, 1993, Proceedings, Lecture Notes in Computer Science, Vol. 707, Springer, Berlin, 1993, pp. 220–246.
- [222] J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96*, Asilomar, CA, September 3–6, 1996, *Electronic Notes in Theoretical Computer Science*, Vol. 4, Elsevier, Amsterdam, 1996. <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [223] J. Meseguer, Rewriting logic as a semantic framework for concurrency: a progress report, in: U. Montanari, V. Sassone (Eds.), *CONCUR'96: Concurrency Theory*, 7th Internat. Conf., Pisa, Italy, August 26–29, 1996, Proceedings, Lecture Notes in Computer Science, Vol. 1119, Springer, Berlin, 1996, pp. 331–372.
- [224] J. Meseguer, Formal interoperability, in: *Proc. of the 1998 Conference on Mathematics in Artificial Intelligence*, Fort Lauderdale, Florida, January, 1998, <http://rutcor.rutgers.edu/~amai/Proceedings.html>. Presented also at the 14th IMACS World Congress, Atlanta, Georgia, July, 1994.
- [225] J. Meseguer, A logical framework for distributed systems and communication protocols, in: S. Budkowski, A. Cavalli, E. Najm (Eds.), *Formal Description Techniques and Protocol Specification, Testing and Verification, FORTE/PSTV'98 IFIP TC6 WG6.1 Joint Internat. Conf. on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE XI) and Protocol Specification, Testing and Verification (PSTV XVIII)*, November 3–6, 1998, Paris, France, International Federation for Information Processing, Vol. 135, Kluwer Academic Publishers, Dordrecht, 1998, pp. 327–333.
- [226] J. Meseguer, Membership algebra as a logical framework for equational specification, in: F. Parisi-Presicce (Ed.), *Recent Trends in Algebraic Development Techniques*, 12th Internat. Workshop, WADT'97, Tarquinia, Italy, June 3–7, 1997, Selected Papers, Lecture Notes in Computer Science, Vol. 1376, Springer, Berlin, 1998, pp. 18–61.
- [227] J. Meseguer, Research directions in rewriting logic, in: U. Berger, H. Schwichtenberg (Eds.), *Computational Logic, Proc. of the NATO Advanced Study Institute on Computational Logic held in Marktoberdorf, Germany, July 29–August 6, 1997*, NATO ASI Series F: Computer and Systems Sciences, Vol. 165, Springer, Berlin, 1998, pp. 347–398.
- [228] J. Meseguer, Rewriting logic and Maude: a wide-spectrum semantic framework for object-based distributed systems, in: S.F. Smith, C.L. Talcott (Eds.), *Proc. IFIP Conf. on Formal Methods for Open Object-Based Distributed Systems IV, FMOODS 2000*, September 6–8, 2000, Stanford, CA, USA, Kluwer Academic Publishers, Dordrecht, 2000, pp. 89–117.
- [229] J. Meseguer, Rewriting logic and Maude: concepts and applications, in: L. Bachmair (Ed.), *Rewriting Techniques and Applications*, 11th Internat. Conf., RTA 2000, Norwich, UK, July 10–12, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1833, Springer, Berlin, 2000, pp. 1–26.
- [230] J. Meseguer, N. Martí-Oliet, From abstract data types to logical frameworks, in: E. Astesiano, G. Reggio, A. Tarlecki (Eds.), *Recent Trends in Data Type Specification*, 10th Workshop on Specification of Abstract Data Types Joint with the 5th COMPASS Workshop, S. Margherita, Italy, May 30–June 3, 1994, Selected Papers, Lecture Notes in Computer Science, Vol. 906, Springer, Berlin, 1995, pp. 48–80.
- [231] J. Meseguer, U. Montanari, Mapping tile logic into rewriting logic, in: F. Parisi-Presicce (Ed.), *Recent Trends in Algebraic Development Techniques*, 12th Internat. Workshop, WADT'97, Tarquinia, Italy, June 3–7, 1997, Selected Papers, Lecture Notes in Computer Science, Vol. 1376, Springer, Berlin, 1998, pp. 62–91.
- [232] J. Meseguer, C.L. Talcott, Using rewriting logic to interoperate architectural description languages (I and II), *Lectures at the Santa Fe and Seattle DARPA-EDCS Workshops*, March and July 1997, <http://www-formal.stanford.edu/clt/ArpaNsf/adl-interop.html>.



- [233] J. Meseguer, C.L. Talcott, Mapping OMRS to rewriting logic, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 345–366, <http://www.elsevier.nl/locate/entcs/volume15.html>. Full version in preparation. Full version in preparation.
- [234] J. Meseguer, C.L. Talcott, A partial order event model for concurrent objects, in: J.C.M. Baeten, S. Mauw (Eds.), CONCUR'99, Concurrency Theory, 10th Internat. Conf. Eindhoven, The Netherlands, August 24–27, 1999, Proceedings, Lecture Notes in Computer Science, Vol. 1664, Springer, Berlin, 1999, pp. 415–430.
- [235] J. Meseguer, T. Winkler, Parallel programming in Maude, in: J.-P. Banâtre, D. Le Métayer (Eds.), Research Directions in High-level Parallel Programming Languages, Mont Saint-Michel, France, June 17–19, 1991, Proceedings, Lecture Notes in Computer Science, Vol. 574, Springer, Berlin, 1992, pp. 253–293.
- [236] J. Meseguer, X. Qian, A logical semantics for object-oriented databases, in: P. Buneman, S. Jajodia (Eds.), Proc. Internat. SIGMOD Conf. on Management of Data, ACM, New York, 1993, pp. 89–98.
- [237] J. Meseguer, K. Futatsugi, T. Winkler, Using rewriting logic to specify, program, integrate, and reuse open concurrent systems of cooperating agents, in: Proc. of the 1992 International Symposium on New Models for Software Architecture, Tokyo, Japan, November 1992, Research Institute of Software Engineering, pp. 61–106.
- [238] J. Meseguer, M.-O. Stehr, C.L. Talcott, Specifying the PLAN language in Maude, 2000, Slides available at <http://www-formal.stanford.edu/clt/Talks/00sep-utokyo-talk.ps.gz>.
- [239] J. Millen, Applications of term rewriting to cryptographic protocol analysis, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 229–234, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [240] H. Miyoshi, Modelling conditional rewriting logic in structured categories. in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 20–34, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [241] U. Montanari, C.L. Talcott, Can actors and pi-agents live together?, in: A.D. Gordon, A.M. Pitts, C.L. Talcott (Eds.), Proc. 2nd Workshop on Higher-Order Operational Techniques in Semantics, HOOFS'98, Electronic Notes in Theoretical Computer Science, Vol. 10, Elsevier, Amsterdam, 1998.
- [242] P.-E. Moreau, A choice-point library for backtrack programming, in: JICSLP'98 Post-Conf. Workshop on Implementation Technologies for Programming Languages based on Logic, June, 1998.
- [243] P.-E. Moreau, Compilation de Règles de Réécriture et de Stratégies Non-Déterministes, Ph.D. Thesis, Université Henri Poincaré — Nancy I, June, 1999.
- [244] P.-E. Moreau, H. Kirchner, Compilation of associative-commutative normalisation with strategies in ELAN (full version), Technical Report 97-R-129, LORIA, Nancy, France, 1997.
- [245] P.-E. Moreau, H. Kirchner, Compilation techniques for associative-commutative normalisation, in: M.P.A. Sellink (Ed.), 2nd Internat. Workshop on the Theory and Practice of Algebraic Specifications, Amsterdam, The Netherlands, September 25–26, 1997, Electronic Workshops in Computing, Springer, Berlin, 1998, <http://www.ewic.org.uk/ewic/workshop/view.cfm/ASFSDF-97>.
- [246] P.-E. Moreau, H. Kirchner, A compiler for rewrite programs in associative-commutative theories, in: C. Palamidessi, H. Glaser, K. Meinke (Eds.), Principles of Declarative Programming, 10th Internat. Symp. PLIP'98 Held Jointly with the 6th Conference ALP'98, Pisa, Italy, September 16–18, 1998, Proceedings, Lecture Notes in Computer Science, Vol. 1490, Springer, Berlin, 1998, pp. 230–249.
- [247] P.D. Mosses, Semantics, modularity, and rewriting logic, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [248] P.D. Mosses, Logical specification of operational semantics, in: J. Flum, M. Rodríguez-Artalejo (Eds.), Computer Science Logic, 13th Internat. Workshop, CSL'99, 8th Ann. Conf. of the EACSL, Madrid, Spain, September 20–25, 1999, Proceedings, Lecture Notes in Computer Science, Vol. 1683, Springer, Berlin, 1999, pp. 32–49.



- [249] E. Najm, J.-B. Stefani, A formal operational semantics for the ODP computational model with signals, explicit binding, and reactive objects, Manuscript, ENST, Paris, France, 1994.
- [250] E. Najm, J.-B. Stefani, A formal semantics for the ODP computational model, *Comput. Networks ISDN Systems* 27 (1995) 1305–1329.
- [251] E. Najm, J.-B. Stefani, Computational models for open distributed systems, in: H. Bowman, J. Derrick (Eds.), *Proc. 2nd IFIP Conf. on Formal Methods for Open Object-Based Distributed Systems, FMOODS'97*, July 21–23, 1997, Canterbury, Kent, UK, Chapman & Hall, London, 1997, pp. 157–176.
- [252] S. Nakajima, Encoding mobility in CafeOBJ: an exercise of describing mobile code-based software architecture, in: K. Futatsugi (Ed.), *Proc. of the CafeOBJ Symposium '98*, Numazu, Japan, CafeOBJ Project, April, 1998.
- [253] S. Nakajima, Using algebraic specification techniques in development of object-oriented frameworks, in: J.M. Wing, J. Woodcock, J. Davies (Eds.), *FM'99 — Formal Methods, World Congress on Formal Methods in the Development of Computing Systems*, Toulouse, France, September 20–24, 1999 Proceedings, Vol. II, *Lecture Notes in Computer Science*, Vol. 1709, Springer, Berlin, 1999, pp. 1664–1683.
- [254] S. Nakajima, K. Futatsugi, An object-oriented modeling method for algebraic specifications in CafeOBJ, in: *Proc. 19th Internat. Conf. on Software Engineering*, Boston, MA, May 1997, IEEE Computer Society Press, Silverspring, MD, pp. 34–44.
- [255] M. Nakamura, K. Ogata, The evaluation strategy for head normal form with and without on-demand flags, in: K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000*, Kanazawa, Japan, September 18–20, 2000, *Electronic Notes in Theoretical Computer Science*, Vol. 36, Elsevier, Amsterdam, 2000, pp. 211–227, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [256] H. Nakashima, Cyber assistance for situated human information processing, in: K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000*, Kanazawa, Japan, September 18–20, 2000, *Electronic Notes in Theoretical Computer Science*, Vol. 36, Elsevier, Amsterdam, 2000, pp. 295–296, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [257] P. Naumov, M.-O. Stehr, J. Meseguer, The HOL/NuPRL proof translator — a practical approach to formal interoperability, in: *Theorem Proving in Higher Order Logics*, 14th Internat. Conf. TPHOLs'2001, Edinburgh, Scotland, UK, September 3–6, 2001, *Proceedings, Lecture Notes in Computer Science*, Springer, Berlin, 2001.
- [258] T. Noll, On coherence properties in term rewriting models of concurrency, in: J.C.M. Baeten, S. Mauw (Eds.), *CONCUR'99, Concurrency Theory*, 10th Internat. Conf. Eindhoven, The Netherlands, August 24–27, 1999, *Proceedings, Lecture Notes in Computer Science*, Vol. 1664, Springer, Berlin, 1999, pp. 478–493.
- [259] M. Numazawa, M. Kurihara, A. Ohuchi, A reflective language based on conditional term rewriting, Technical report, Division of Systems and Information Engineering, Hokkaido University, Sapporo, Japan, 1996.
- [260] K. Ogata, K. Futatsugi, An abstract machine for order-sorted conditional term rewriting systems, in: H. Comon (Ed.), *Rewriting Techniques and Applications*, 8th Internat. Conf., RTA'97, Sitges, Spain, June 2–5, 1997, *Proceedings, Lecture Notes in Computer Science*, Vol. 1232, Springer, Berlin, 1997, pp. 335–338.
- [261] K. Ogata, K. Futatsugi, Specification and verification of some classical mutual exclusion algorithms with CafeOBJ, in: K. Futatsugi, J.A. Goguen, J. Meseguer (Eds.), *OBJ/CafeOBJ/Maude Workshop at Formal Methods '99: Formal Specification, Proof, and Applications*, Theta, Bucharest, 1999, pp. 159–178.
- [262] P.C. Ölveczky, Specification and analysis of real-time and hybrid systems in rewriting logic, Ph.D. Thesis, University of Bergen, Norway, 2000, <http://maude.csl.sri.com/papers>.
- [263] P.C. Ölveczky, S. Meldal, Specification and prototyping of network protocols in rewriting logic, in: *Proc. of NIK'98, Norsk Informatikk Konferanse*, 1998.
- [264] P.C. Ölveczky, J. Meseguer, Specifying real-time systems in rewriting logic, in: J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96*, Asilomar, CA, September 3–6, 1996, *Electronic Notes in Theoretical Computer Science*, Vol. 4, Elsevier, Amsterdam, 1996, pp. 283–308, <http://www.elsevier.nl/locate/entcs/volume4.html>.

- [265] P.C. Ölveczky, J. Meseguer, Real-time Maude: a tool for simulating and analyzing real-time and hybrid systems, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam 2000, pp. 361–383, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [266] P.C. Ölveczky, J. Meseguer, Specification of real-time and hybrid systems in rewriting logic, Theoret. Comput. Sci. (2002), submitted for publication.
- [267] P.C. Ölveczky, J. Meseguer, Specifying and analyzing real-time object systems in real-time Maude, Manuscript, Computer Science Laboratory, SRI International, 2001, submitted for publication.
- [268] P.C. Ölveczky, M. Keaton, J. Meseguer, C.L. Talcott, S. Zabele, Specification and analysis of the AER/NCA active network protocol suite in Real-Time Maude, in: H. Hussmann (Ed.), Fundamental Approaches to Software Engineering, 4th Internat. Conf., FASE 2001, Held as Part of ETAPS 2001, Genova, Italy, April 2001, Proceedings, Lecture Notes in Computer Science, Vol. 2029, Springer, Berlin, 2001, pp. 333–347. <http://maude.csl.sri.com/papers>.
- [269] P.C. Ölveczky, P. Kosiuczenko, M. Wirsing, An object-oriented algebraic steam-boiler control specification, in: J.-R. Abrial, E. Börger, H. Langmaack (Eds.), Formal Methods for Industrial Applications: Specifying and Programming the Steam Boiler Control, Lecture Notes in Computer Science, Vol. 1165, Springer, Berlin, 1996, pp. 379–402.
- [270] D. Pattinson, Modal logic for rewriting theories, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam 2000, pp. 173–191, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [271] I. Pita, N. Martí-Oliet, A Maude specification of an object oriented database model for telecommunication networks, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 404–422, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [272] I. Pita, N. Martí-Oliet, Using reflection to specify transaction sequences in rewriting logic, in: J.L. Fiadeiro (Ed.), Recent Trends in Algebraic Development Techniques, 13th Internat. Workshop, WADT'98, Lisbon, Portugal, April 2–4, 1998, Selected Papers, Lecture Notes in Computer Science, Vol. 1589, Springer, Berlin, 1999, pp. 261–276.
- [273] I. Pita, N. Martí-Oliet, A Maude specification of an object-oriented model for telecommunication networks, Theoret. Comput. Sci. 2002, submitted for publication.
- [274] J.F. Quesada, The Maude parser: parsing and meta-parsing  $\beta$ -extended context-free grammars, Technical report, Computer Science Laboratory, SRI International, 2001, to appear.
- [275] C. Ringeissen, Prototyping combination of unification algorithms with the ELAN rule-based programming language, in: H. Comon (Ed.), Rewriting Techniques and Applications, 8th Internat. Conf. RTA'97, Sitges, Spain, June 2–5, 1997, Proceedings, Lecture Notes in Computer Science, Vol. 1232, Springer, Berlin, 1997, pp. 323–326.
- [276] C. Ringeissen, Handling relations over finite domains in the rule-based system ELAN, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam 2000, pp. 193–210, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [277] D.E. Rodríguez, Case studies in the specification and analysis of protocols in Maude, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam 2000, pp. 257–275, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [278] G. Roşu, K. Havelund, Generating optimal monitors from temporal formulae, Technical report, Research Institute for Advanced Computer Science, 2001.
- [279] B. Salmansberger, Objektorientierte Spezifikation von verteilten Systemen in Maude am Beispiel eines Flughafens, Master's Thesis, Fakultät für Mathematik und Informatik, Universität Passau, December, 1993.
- [280] M. Sánchez, J.L. Herrero, J.M. Murillo, J. Hernández, Guaranteeing coherent software systems when composing coordinated components, in: A. Porto, G.-C. Roman (Eds.), Coordination Languages and

- Models, 4th Internat. Conf. COORDINATION 2000, Limassol, Cyprus, September 11–13, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1906, Springer, Berlin, 2000, pp. 341–346.
- [281] C. Scharff, *Déduction avec Contraintes et Simplification dans les Théories Équationnelles*, Ph.D. Thesis, Université Henri Poincaré — Nancy I, September, 1999.
- [282] W.M. Schorlemmer, Bi-rewriting rewriting logic, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 265–282, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [283] W.M. Schorlemmer, Rewriting logic as a logic of special relations, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 163–184. <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [284] W.M. Schorlemmer, On specifying and reasoning with special relations, Ph.D. Thesis, Universitat Politècnica de Catalunya, March, 1999.
- [285] W.M. Schorlemmer, Term rewriting in a logic of special relations, in: A.M. Haeberer (Ed.), Algebraic Methodology and Software Technology, 7th Internat. Conf. AMAST'98, Amazonia, Brazil, January 4–8, 1999, Proceedings, Lecture Notes in Computer Science, Vol. 1548, Springer, Berlin, 1999, pp. 178–195.
- [286] D.B. Skillicorn, D. Talia, Models and languages for parallel computation, ACM Comput. Surveys 30 (2) (1998) 123–169.
- [287] L.J. Steggle, Rewriting logic and Elan: prototyping tools for Petri nets with time, in: J.-M. Colom, M. Koutny (Eds.), Applications and Theory of Petri Nets 2001, 22nd Internat. Conf. ICATPN 2001, Newcastle upon Tyne, UK, June 25–29, 2001, Proceedings, Lecture Notes in Computer Science, Vol. 2075, Springer, Berlin, 2001, pp. 363–381.
- [288] L.J. Steggle, P. Kosiuczenko, A timed rewriting logic semantics for SDL: a case study of the alternating bit protocol, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 295–316, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [289] L.J. Steggle, P. Kosiuczenko, A formal model for SDL specifications based on timed rewriting logic, Automat. Software Eng. 7 (1) (2000) 61–90.
- [290] M.-O. Stehr, CINNI — A generic calculus of explicit substitutions and its application to  $\lambda$ -,  $\zeta$ - and  $\pi$ -calculi, in: K. Futatsugi (Ed.), Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam 2000, pp. 71–92, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [291] M.-O. Stehr, A rewriting semantics for algebraic nets, in: C. Girault, R. Valk (Eds.), Petri Nets for System Engineering — A Guide to Modeling, Verification, and Applications, Springer, Berlin, 2001.
- [292] M.-O. Stehr, Rewriting logic and type theory — from applications to unification, Ph.D. Thesis, Computer Science Department, University of Hamburg, Germany, 2002, in preparation.
- [293] M.-O. Stehr, J. Meseguer, Pure type systems in rewriting logic, in: Proc. of LFM'99: Workshop on Logical Frameworks and Meta-languages, Paris, France, September, 28, 1999, <http://www.cs.bell-labs.com/~felty/LFM99/>.
- [294] M.-O. Stehr, J. Meseguer, P.C. Ölveczky, Representation and execution of Petri nets using rewriting logic as a uniform framework, in: J. Padberg (Ed.), Proc. UNIGRA'2001, Uniform Approaches to Graphical Process Specification Techniques, Electronic Notes in Theoretical Computer Science, Genova, Italy, March/April 2001, Elsevier, Amsterdam, to appear.
- [295] M.-O. Stehr, J. Meseguer, P.C. Ölveczky, Rewriting logic as a unifying framework for Petri nets, in: H. Ehrig, G. Juhas, J. Padberg, G. Rozenberg (Eds.), Unifying Petri Nets, Lecture Notes in Computer Science, Springer, Berlin, 2001, to appear.
- [296] M.-O. Stehr, P. Naumov, J. Meseguer, A proof-theoretic approach to the HOL-Nuprl connection with applications to proof translation, in: M. Cerioli, P.D. Mosses, G. Reggio (Eds.), Proc. WADT/CoFI'01, 15th Internat. Workshop on Algebraic Development Techniques and General Workshop of the CoFI WG, Genova, Italy, April 1–3, 2001, <http://www.csl.sri.com/~stehr/fi.eng.html>.

- [297] Y. Tahara, F. Kumeno, A. Ohsuga, S. Honiden, An algebraic semantics of reflective objects, in: K. Futatsugi, S. Matsuoka (Eds.), *Object-Technologies for Advanced Software*, Second JSSST Internat. Symp. ISOTAS'96, Kanaza, Japan, March 11–15, 1996, Proceedings, Lecture Notes in Computer Science, Vol. 1049, Springer, Berlin, 1996, pp. 173–189.
- [298] C.L. Talcott, An actor rewriting theory, in: J. Meseguer (Ed.), *Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96*, Asilomar, CA, September 3–6, 1996, *Electronic Notes in Theoretical Computer Science*, Vol. 4, Elsevier, Amsterdam, 1996, pp. 360–383, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [299] C.L. Talcott, Composable semantic models for actor theories, in: M. Abadi, T. Ito (Eds.), *Theoretical Aspects of Computer Science*, 3rd Internat. Symp. TACS'97, Sendai, Japan, September 23–26, 1997, Proceedings, Lecture Notes in Computer Science, Vol. 1281, Springer, Berlin, 1997, pp. 321–364.
- [300] C.L. Talcott, Interaction semantics for components of distributed systems, in: E. Najm, J.-B. Stefani (Eds.), *Proc. IFIP Conf. on Formal Methods for Open Object-Based Distributed Systems, FMOODS'96*, Chapman & Hall, London, 1997, pp. 154–169.
- [301] C.L. Talcott, Towards a toolkit for actor system specification, in: T. Rus (Ed.), *Algebraic Methodology and Software Technology*, 8th Internat. Conf. AMAST 2000, Iowa City, Iowa, USA, May 20–27, 2000, Proceedings, Lecture Notes in Computer Science, Vol. 1816, Springer, Berlin, 2000, pp. 391–406.
- [302] C.L. Talcott, Actor theories in rewriting logic, *Theoret. Comput. Sci.* (2001), submitted for publication.
- [303] M.P. Tarjuelo, Comparing Meseguer's rewriting logic with the logic CRWL, in: *Proc. WFLP 2001*, Internat. Workshop on Functional and (Constraint) Logic Programming, Kiel, Germany, September 13–15, 2001.
- [304] M.P. Tarjuelo, Relating Meseguer's rewriting logic with the constructor-based rewriting logic, Master's Thesis, Facultad de Matemáticas, Universidad Complutense de Madrid, Spain, May, 2001, <http://maude.csl.sri.com/papers>.
- [305] A. Toval, J.L. Fernández, Formally modeling UML and its evolution: a holistic approach, in: S.F. Smith, C.L. Talcott (Eds.), *Proc. IFIP Conf. on Formal Methods for Open Object-Based Distributed Systems IV, FMOODS 2000*, September 6–8, 2000, Stanford, CA, USA, Kluwer Academic Publishers, Dordrecht, 2000, pp. 183–206.
- [306] A. Toval, J.L. Fernández, Improving system reliability via rigorous software modeling: the UML case, in: *Proc. IEEE Aerospace Conf. Vol. 6, Big Sky, MT, USA, March 10–17, 2001*, IEEE Press, New York, pp. 6–17.
- [307] M.G.J. van den Brand, P. Klint, C. Verhoef, Term rewriting for sale, in: C. Kirchner, H. Kirchner (Eds.), *Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98*, Pont-à-Mousson, France, September 1–4, 1998, *Electronic Notes in Theoretical Computer Science*, Vol. 15, Elsevier, Amsterdam, 1998, pp. 139–162, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [308] M.G.J. van den Brand, C. Ringeissen, ASF+SDF parsing tools applied to ELAN, in: K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA 2000*, Kanazawa, Japan, September 18–20, 2000, *Electronic Notes in Theoretical Computer Science*, Vol. 36, Elsevier, Amsterdam, 2000, pp. 135–154, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [309] A. Verdejo, N. Martí-Oliet, Executing E-LOTOS processes in Maude, in: H. Ehrig, M. Grosse-Rhode, F. Orejas (Eds.), *INT 2000, Integration of Specification Techniques with Applications in Engineering*, Extended Abstracts, Technical report 2000/04, Technische Universität Berlin, March, 2000, pp. 49–53.
- [310] A. Verdejo, N. Martí-Oliet, Executing and verifying CCS in Maude, Technical Report 99-00, Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, February 2000, <http://maude.csl.sri.com/casestudies/ccs>.
- [311] A. Verdejo, N. Martí-Oliet, Implementing CCS in Maude, in: T. Bolognesi, D. Latella (Eds.), *Formal Methods For Distributed System Development, FORTE/PSTV 2000 IFIP TC6 WG6.1 Joint Internat. Conf. on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE XIII) and Protocol Specification, Testing and Verification (PSTV XX)* October 10–13, 2000, Pisa, Italy, International Federation for Information Processing, Vol. 183, Kluwer Academic Publishers, Dordrecht, 2000, pp. 351–366.
- [312] A. Verdejo, I. Pita, N. Martí-Oliet, The leader election protocol of IEEE 1394 in Maude, in: K. Futatsugi (Ed.), *Proc. 3rd Internat. Workshop on Rewriting Logic and its Applications, WRLA*

- 2000, Kanazawa, Japan, September 18–20, 2000, Electronic Notes in Theoretical Computer Science, Vol. 36, Elsevier, Amsterdam, 2000, pp. 385–406, <http://www.elsevier.nl/locate/entcs/volume36.html>.
- [313] P. Viry, La réécriture concurrente, Ph.D. Thesis, Université de Nancy I, 1992.
- [314] P. Viry, Rewriting: an effective model of concurrency, in: C. Halatsis, D. Maritsas, G. Philokyprou, S. Theodoridis (Eds.), PARLE'94 Parallel Architectures and Languages Europe, 6th Internat. PARLE Conf. Athens, Greece, July 4–8, 1994, Proceedings, Lecture Notes in Computer Science, Vol. 817, Springer, Berlin, 1994, pp. 648–660.
- [315] P. Viry, Rewriting modulo a rewrite system, Technical Report TR-95-20, Dipartimento di Informatica, Università di Pisa, December, 1995, <ftp://ftp.di.unipi.it/pub/techreports/TR-95-20.ps.Z>.
- [316] P. Viry, Input/output for ELAN, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 51–64, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [317] P. Viry, Adventures in sequent calculus modulo equations, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 367–378, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [318] P. Viry, Equational rules for rewriting logic, Theoret. Comput. Sci. (2002), submitted for publication.
- [319] E. Visser, Z. el A. Benaissa, A core language for rewriting, in: C. Kirchner, H. Kirchner (Eds.), Proc. 2nd Internat. Workshop on Rewriting Logic and its Applications, WRLA'98, Pont-à-Mousson, France, September 1–4, 1998, Electronic Notes in Theoretical Computer Science, Vol. 15, Elsevier, Amsterdam, 1998, pp. 25–44, <http://www.elsevier.nl/locate/entcs/volume15.html>.
- [320] M. Vittek, ELAN: Un Cadre Logique pour le Prototypage de Langages de Programmation avec Contraintes, Ph.D. Thesis, Université Henri Poincaré — Nancy I, November, 1994.
- [321] M. Vittek, A compiler for nondeterministic term rewriting systems, in: H. Ganzinger (Ed.), Rewriting Techniques and Applications, 7th Internat. Conf. RTA'96, New Brunswick, NJ, USA July 27–30, 1996, Proceedings, Lecture Notes in Computer Science, Vol. 1103, Springer, Berlin, 1996, pp. 154–168.
- [322] B.-Y. Wang, J. Meseguer, C.A. Gunter, Specification and formal analysis of a PLAN algorithm in Maude, in: P.-A. Hsiung (Ed.), Proc. Internat. Workshop on Distributed System Validation and Verification, Taipei, Taiwan, April, 2000, pp. 49–56.
- [323] T. Watanabe, Towards a foundation of computational reflection based on abstract rewriting (preliminary result), in: Proc. IMSA'95, Information-Technology Promotion Agency, Japan, 1995, pp. 143–145.
- [324] T. Watanabe, H. Ishikawa, K. Futatsugi, Towards declarative description of computational reflection, in: Proc. IMSA'96, Information-Technology Promotion Agency, Japan, 1996, pp. 113–128.
- [325] T. Winkler, Programming in OBJ and Maude, in: P. Lauer (Ed.), Functional Programming, Concurrency, Simulation and Automated Reasoning, International Lecture Series 1991–1992, McMaster University, Hamilton, Ontario, Canada, Lecture Notes in Computer Science, Vol. 693, Springer, Berlin, 1993, pp. 229–277.
- [326] M. Wirsing, A. Knapp, A formal approach to object-oriented software engineering, in: J. Meseguer (Ed.), Proc. 1st Internat. Workshop on Rewriting Logic and its Applications, WRLA'96, Asilomar, CA, September 3–6, 1996, Electronic Notes in Theoretical Computer Science, Vol. 4, Elsevier, Amsterdam, 1996, pp. 321–359, <http://www.elsevier.nl/locate/entcs/volume4.html>.
- [327] M. Wirsing, A. Knapp, A formal approach to object-oriented software engineering, Theoret. Comput. Sci. (2002), submitted for publication.
- [328] M. Wirsing, F. Nickl, U. Lechner, Concurrent object-oriented specification in SPECTRUM, in: Y. Inagaki (Ed.), Workshop on Algebraic and Object-Oriented Approaches to Software Science, Nagoya, Japan, 1995, pp. 39–70.