



Project IST-2001-38314, COLUMBUS

Design of Embedded Controllers for Safety Critical Systems

Deliverable DTA2

Report on formal framework of meta-models – resp. INRIA

Reference Period: 31 December 2003 – 30 June 2004

June 23, 2004

Project Co-ordinator

Organisation: *Department of Electrical and Computer Engineering
University of Cambridge*

Responsible person: *Dr John Lygeros*

Address: *University Campus
Rio, Patras, GR 26500, Greece*

Phone: *+30 2610 996458*

Fax: *+30 2610 991812*

E-mail: *lygeros@ee.upatras.gr*

Consortium

<i>Participant name</i>	<i>Acronym</i>	<i>Role</i>
1 <i>University of Patras</i>	<i>UPAT</i>	<i>Coordinator</i>
2 <i>University of Cambridge</i>	<i>UCAM</i>	<i>Contractorr</i>
3 <i>University of l'Aquila</i>	<i>AQUI</i>	<i>Contractor</i>
4 <i>Institut National de Recherche en Informatique et en Automatique</i>	<i>INRIA</i>	<i>Contractor</i>
5 <i>University of California, Berkeley</i>	<i>UCB</i>	<i>Contractor</i>
6 <i>Vanderbilt University</i>	<i>VU</i>	<i>Contractor</i>
7 <i>PARADES, Rome</i>	<i>PARADES</i>	<i>Subcontractor</i>

June 30, 2004

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
0.1	23/06/2004	First draft	Draft	Partners
0.2	30/06/2004		Final	Partners

June 30, 2004

TABLE of CONTENTS

1.	LIST OF DUE DELIVERABLES FOR WPTA	4
2.	SUMMARY OF RESULTS	5
2.1	Recalling the objectives	5
2.2	It is possible to optimize across MoCC boundaries to improve performance and reduce errors in the design at early stages in the process?	5
2.3	Meta-modeling techniques for heterogeneous modeling	7
2.4	References	8
3.	MEETINGS, VISITS, EXCHANGES OF PEOPLE	10
3.1	Meetings	10
3.2	Visits	10
4.	ANNEXE	10

1. LIST OF DUE DELIVERABLES FOR WPTA

Id	Deliverable	Respons. Partner	Original due date/ milestone	Revised due date/ milestone	Actual delivery	Status / Comments
DTA1	Report on formal framework of meta-models	INRIA	31/12/03		23/01/04	
DTA2	Report on use of meta-model for hybrid systems.	INRIA	30/6/04		30/06/04	
DTA3	Contribution to consolidated final project report	INRIA	30/6/04		30/06/04	

2. SUMMARY OF RESULTS

2.1 Recalling the objectives

To serve as a reference for assessing our results, we first quote here selected key sentences of the project document regarding this work package:

Execution environments and development environments are two essential components of embedded systems development. Meta models and meta modeling play a crucial role in both areas. In execution environments, meta modeling is used for capturing the semantics of models of computations and communications (MoCC). In development environments meta modeling offers an effective approach to formally model the abstract syntax and static semantics of domain-specific modeling languages. In this work package we will contribute to the development of advanced meta modeling techniques in both directions as described below.

- 1. An essential aspect of the research is the development of formal definition of the semantics of communications so that implementation choices will be correct by construction. Several formal models have been proposed over the years to capture one or more aspects of computations using a unifying theoretical framework introduced recently by Lee and Sangiovanni-Vincentelli. However, this denotational framework has only helped us to identify the sources of difficulties in combining different MoCC that are certainly needed when complex systems are being designed. [...] We believe that it is possible to optimize across MoCC boundaries to improve performance and reduce errors in the design at early stages in the process.*
- 2. Domain-specific modeling languages have significant impact on the design process. In embedded systems, where communication and computation always occur in the context of physical domain, domain-specific languages offer an effective way to structure the information about the system to be designed along the “natural dimensions” of the applications. [...] In this work package, we will extend previous work on the use of meta-models for the specification and validation of modeling languages for hybrid systems. Of particular interest are mathematical formalisms that enable the concise representation and composition of complex, multiple-aspect modeling languages.*

In comparison with the above stated objectives we state below a summary of what has been achieved at this stage of the project.

2.2 It is possible to optimize across MoCC boundaries to improve performance and reduce errors in the design at early stages in the process?

This paragraph summarizes the results obtained regarding the above point 1 quoted from the project document. Corresponding results have been mainly obtained through a deep, tight, and continuing cooperation between researchers from INRIA (Albert Benveniste, Benoît Caillaud, and Dumitru Potop-Butucaru), PARADES (Alberto Sangiovanni-Vincentelli), and UCB (Luca Carloni and ASV) – in addition, we have invited Paul Caspi, from VERIMAG, to join our group for this work. Main results are listed now.

2.2.1 Theory of heterogeneous modeling

Results. We have made key progresses in the *theory of heterogeneous modeling*, the theory dealing with how to give a precise semantics to the assembly of models based on different MoCCs, and how to map a given design onto an architecture with a different MoCC. To this end, as planned, we have started from the Lee and Sangiovanni-Vincentelli (LSV) tagged systems model.

- We have shown how to adapt this model with the objective of regarding the MoCC as a parameter. This being done, we were able to define *morphisms relating different MoCCs*, i.e., classes of mappings between different MoCCs, satisfying certain properties. Roughly speaking, morphisms formalize the generic concept of relaxing synchronization – it can be by

June 30, 2004

removing synchrony, by losing causality information, or by removing real-time information. Morphisms compose.

- With this at hand, we could define *heterogeneous parallel composition*, a composition that provides formal meaning to the assembly of models with different MoCCs.
- We could formally state what it means to preserve the semantics when deploying a design over an architecture not complying with the original MoCC of the design.
- Finally we got a first set of theorems characterizing when such a deployment does preserve the semantics. This formalizes the notion of “correct-by-construction” deployment.

New progresses since DTA1. We have now completed the picture of our *theory of heterogeneous modeling*, by adding to it the following features:

- We have extended our former parallel composition by intersection, by allowing more general parallel compositions. Our more general framework allows capturing causality and scheduling constraints, compositional performance evaluation models (Worst Case Execution Times), something our previous approach was not able to encompass.
- We have formally defined *heterogeneous architectures*. This required non trivial efforts, well beyond our results from DTA1 where we did not seriously investigate the case where more than 2 components are involved. The problem is with the lack of associativity of heterogeneous parallel composition – heterogeneous parallel composition cannot be associative. To cope with this difficulty and still get a meaningful definition for what $P // P' // P''$ should be, we had to use pullbacks from category theory, something we did not expect when starting this work.
- We have discovered a “separation principle” for correct-by-construction deployment, which says the following: the problem of deploying a heterogeneous design over a (possibly different) heterogeneous architecture can be split into two distinct problems, the one concerning abstract synchronization mechanisms between the different software components, and the other one consisting in ensuring that some “desynchronized” version of the communication medium behaves like an asynchronous identity medium.
- We have applied the entire body of results to formally prove the correctness of the LTTA-based design methodology in use at some aircraft manufacturers.

Perspectives. The subsequent steps of this direction of work will be twofold:

- Making the above results more effective. The theorems obtained are not enough yet to derive effective algorithms for correct-by-construction deployment; the reason is that they deal with models of traces, i.e., infinite behaviours, not effective machines. Our next effort will consist in giving an effective variant of the LSV model, not for trace models, but for effective machines.
- Taking advantage of the skills collected at the COLUMBUS team, we want to enhance our tagged systems models with probabilities. This would allow us to address correct-by-construction deployment in combination with fault-tolerance.

These research directions will be pursued outside COLUMBUS.

2.2.2 A framework to compare desynchronization and latency-insensitive design

Results. L. Carloni and A. Sangiovanni-Vincentelli have used the LSV model as a common framework to offer a comparative exposition of various design approaches: synchronous, asynchronous, GALS, latency-insensitive, and synchronous programming [6,7]. In particular, they studied the interplay among the concepts of event absence, event sampling, and communication latency in modeling distributed heterogeneous design. Furthermore, they presented a new comparison of synchronous program desynchronization and latency-insensitive design. The main operational difference between latency-insensitive design and synchronous program desynchronization can be expressed as follows: the former knows how to handle *black box* processes but does not know how to analyze/exploit *white box* ones (that are treated uniformly as if they were black box processes); the latter does not know how to handle black box processes and must analyze the inner structure of each

June 30, 2004

white box process in the system (as well as the properties of each communicating pair), but it is clever in exploiting the information resulting from this analysis.

New progresses since DTA1. We have investigated how to apply the knowledge of the inner structure of white box processes to derive optimized latency-insensitive protocols. For instance, in the case of a system specified as a network of communicating finite state machines (FSM), we can analyze separately the state-transition function of each FSM to detect the conditions under which such function is sensitive to changes in the values of the FSM input variables. Then, for each state of the FSM we can derive a subset of the input variables that do not affect the function (*don't care* variable for the given state) and use this information to absorb possible incoming *stalling events* on this variables, thereby avoiding to stall unnecessarily the FSM.

Perspectives. The subsequent step in this direction is to design a new family of *shell modules* implementing a latency-insensitive protocol that is able dynamically to detect and use the information on functional-independence in the state-transition function. The main challenge here is to minimize implementation overhead, e.g. area and delay of the shell logic in case of hardware systems.

An additional avenue of future research is centered on designing new latency-insensitive protocols that accommodate also support for fault-tolerant communication mechanisms. The original motivation of latency-insensitive design was to make designs robust with respect to arbitrary variation in communication latency among system components. Here the goal would be to include robustness with respect to transient faults at the component level while maintaining the compositional nature of the original approach.

2.2.3 Improvements on the theory of desynchronization

BENOIT TO UPDATE WHAT FOLLOWS

Results. Dumitru Potop-Butucaru (the post-doc at INRIA working in part for COLUMBUS) and Benoît Caillaud have together discovered a flaw in the 2000 paper by Benveniste, Caillaud and Le Guernic that introduced the new concepts of endochrony and isochrony [3]. The flaw does not concern the central results, but still impairs the satisfactory handling of GALS architectures (GALS with more than 2 components are not covered!). In their quest for correcting this, they have found an important weakening of the two concepts of endochrony and isochrony, respectively called *weak-endochrony* and *weak-isochrony*. These new notions adequately address the problems due to possible internal concurrency inside the components. This is a major result that has been submitted for publication [4]. This paper provides effective criteria and algorithms for the synthesis of correct-by-construction deployment of synchronous designs over GALS architectures.

Perspectives. The follow-up of this work will join the former track. Techniques developed here will be in fact reused to address the case of tag machines, as mentioned in the previous bullet. Thus these two lines of work nicely complement each other, and open the route to a fully general algorithmic treatment of heterogeneity, for both modeling and deployment.

2.2.4 Concluding comments

This line of research conveniently addresses heterogeneous systems made of subsystems with drastically different MoCCs, e.g.: synchronous, asynchronous, with/without causality, with/without scheduling, timed/untimed, etc. *It is our opinion that, in this direction, we have achieved much more than was expected at the writing of the proposal. In addition, the results we have obtained so far and the new directions for research that emerge are even more promising and will have to be pursued beyond the end of the COLUMBUS contract. This line of research is expected to significantly contribute to the fundamentals of platform-based design.*

2.3 Meta-modeling techniques for heterogeneous modeling

BENOIT AND JANOS TO UPDATE WHAT FOLLOWS

Results. The research track presented in section 2.2 allows us to handle MoCCs that are “simple enough” to formulate so they can be entirely captured by a high-level mathematical language. With the advantage that non-trivial theorems can be derived, this is the interest of the approach developed in section 2.2.

June 30, 2004

However, not every MoCC is simple enough to be captured and analysed in this way. For example, the “more than 20 different semantics of Statecharts” won’t be distinguished by high-level mathematical features, but rather by little details in the way actions, transitions, micro-steps, steps, and super-steps, are constructed to form the reactions. Such variants are much better described by means of detailed expansions of each MoCC in terms of a common, lower level but more tunable, common semantic basis. This is referred to as the technique of *compositional metamodeling*. It is developed in the context of a metaprogrammable tool suite for model-based design at the VU group. The metamodels, representing the abstract syntax of Domain Specific Modeling Languages (DSML) are expressed as UML class diagram and the Object Constraint Language (OCL). (Currently, the OMG standard MOF – Meta Object Facility - is adopted as metalanguage.) Compositional metamodeling is used the following way.

1. Elementary behavioural semantics concepts are captured by simple DSML-s.
2. The abstract syntax of these DSML-s are represented by metamodels
3. Metamodel composition techniques – supported by the ISIS-VU GME tool suite – are used to compose DSML-s with complex behavioural semantics from these basic building blocks. (<http://www.isis.vanderbilt.edu/Projects/gme/default.html>)

This is a convenient approach if the difficulty of the considered semantics lies in the combinatorial complexity of the details of the different concepts, not in their high-level mathematical nature.

This aspect of the problem has been the subject of a joint VU-INRIA investigation performed by Ethan Jackson, from VU. The problem considered was to see how meta-modeling techniques can be used to assist the Signal programmer¹ in developing “safe” programs, i.e., programs that will not be problematic from the point of view of synchronization and clocks. This is an interesting application since it concerns a formalism that is quite different from the more classical imperative, state-based, formalisms such as the Statecharts. To this end, a new DSML with two different aspects has been proposed. One aspect concerns data-flow specifications, and the other one addresses synchronization constraints using Signal clocks. Expressing these two aspects as elementary DSML-s (with their own internal abstract syntax and semantics) and composing those using GME composition operators, we expect to gain a multiple-aspect approach for creating Signal programs.

Perspectives. Thanks to compositionality of the Signal language as well as the services included in the Polychrony tool for Signal, specifications involving the two different aspects can be composed and jointly analyzed. The next questions are: (1) Given specifications derived using *different* tools or notations, e.g., Signal and Statecharts, how can GME be used to facilitate the integration of these specifications? (2) How to support the more complex inter-dependence among the two modelling aspects using advanced model transformation capabilities?

2.4 References

- [1] A. Benveniste, B. Caillaud, L. Carloni, P. Caspi, A. Sangiovanni-Vincentelli. **Heterogeneous Reactive Systems Modeling and Correct-by-Construction Deployment: extended case.** To appear in post-publication of the FMCO’2003 conference, LNCS.
- [2] D. Potop-Butucaru, B. Caillaud, and A. Benveniste. **Concurrency in Synchronous systems.** Research Report INRIA, No 5110, February 2004. Also submitted for publication. Dec. 2003.
- [3] D. Potop-Butucaru, B. Caillaud, A. Benveniste, **Concurrency in Synchronous Systems, in Proceedings of the International Conference on Application of Concurrency to System Design, ACSD 2004, 2004.**
- [4] A. Benveniste, B. Caillaud, L. Carloni, P. Caspi, A. Sangiovanni-Vincentelli, **Heterogeneous Reactive Systems Modeling: Capturing Causality and the Correctness of Loosely Time-**

¹ Signal is a synchronous language that has been invented and developed at INRIA-Rennes by Paul Le Guernic and Albert Benveniste. A commercial version of it exists and has been developed and is marketed by TNI-Valiosys, a company located in France.

June 30, 2004

Triggered Architectures (LTTA), in *Embedded Software, Fourth International Workshop, EMSOFT 2004*, G. Buttazzo, S. Edwards (eds.), Pisa, Italy, September 2004.

[5] A. Benveniste, B. Caillaud, L. Carloni, P. Caspi, A. Sangiovanni-Vincentelli, **Composing Heterogeneous Reactive Systems**. Submitted for publication, to appear as an INRIA Research Report, July 2004.

[6] L. P. Carloni and A. L. Sangiovanni-Vincentelli, **A Formal Modeling Framework for Deploying Synchronous Designs on Distributed Architectures**. In FMGALS 2003: First Intl. Workshop on Formal Methods for Globally Asynchronous Locally Synchronous Architectures, September 2003.

[7] L. P. Carloni and A. L. Sangiovanni-Vincentelli, **A Framework for Heterogeneous Systems Modeling and Distributed Deployment of Synchronous Designs**. Submitted to Formal Methods in System Design – An International Journal, 2004.

BENOIT TO COMPLETE

June 30, 2004

3. MEETINGS, VISITS, EXCHANGES OF PEOPLE

3.1 Meetings

The following meetings relevant to the WPTA were held during the considered period.

A COLUMBUS progress meeting was held on March 1-2, 2004, in PARADES, Rome, Italy. This meeting was particularly fruitful in fixing the details of the EMSOFT'04 results and ACM journal submission. Also, it was discovered that fruitful interaction could emerge between WPTA and WPSHS for future common research.

ALBERTO BENOIT TO COMPLETE

3.2 Visits

ALBERTO BENOIT TO COMPLETE

1. Visit of Ethan Jackson, graduate student at ISIS-VU, at INRIA in October, 2003
2. Visit of Ethan Jackson, graduate student at ISIS-VU at INRIA in July, 2004

4. ANNEXE

The following document is considered as the central one, and is therefore integral part of this report:

- Ref. [5] above, which was invited for the special issue of *ACM Transactions on Embedded Systems* devoted to selected papers from EMSOFT'03.