Component Based System Modelling
- Research Plan -

Aneta Vulgarakis

Department of Computer Science and Electronics Mälardalen University,
The PROGRESS Centre for Predictable Embedded Software Systems

aneta.vulgarakis@mdh.se
# Table of contents

1. Research Area ................................................................. 1

2. Related Work ...................................................................... 2
   2.1. Generic Component Models ........................................... 2
   2.2. Functional Property Refinement ..................................... 4

3. Central literature and Seminal Papers ................................. 6

4. Conferences and Leading Research Groups ......................... 7
   4.1. Key Conferences .......................................................... 7
   4.2. Leading Research Groups and People ............................. 8

5. Research issues ............................................................... 9
   5.1. Detailed Description ..................................................... 9
   5.2. Research Method ........................................................ 12
   5.3. Research Questions .................................................. 12
   5.4. Hypothesis ............................................................... 12
   5.5. Expected Results ....................................................... 12

6. Activities ................................................................. 13
   6.1. Courses ................................................................. 13
   6.2. Planned Publications ............................................... 13
   6.3. Corporation ............................................................ 14
   6.4. Conference Trips ...................................................... 14

7. Time plane and Milestones ................................................. 15

8. References .......................................................... 17
1 Research Area

My research will be partitioned between the two interrelated research directions within the Component-based Development (CBD) Cluster: Component-based Software Engineering (CBSE) and Component Verification.

Component-based system development proceeds by assembling pre-existing software components. Component-based software development can significantly reduce development cost and time-to-market, and improve reliability, maintainability and overall quality of software systems.

In recent years, embedded systems have not been the primary target of CBSE. The CBD’s cluster aim is to increase the research excellence in the area of component based software modelling and design for embedded real-time systems. Furthermore, in classic engineering disciplines a component is a self-contained part or subsystem that can be used as a building block in the design of a larger system. On the other hand, in CBSE, there are many different suggestions for a precise definition of a component. One possible view of a component is a blackbox whose implementation is visible only through its interfaces. However, because it’s a subject to third party composition a component’s interface must be specified. Accordingly, current CBD lacks of universally accepted component definition.

One of the proposed methods for writing requirements specification are the formal methods. With the formal specification it is possible to shape a precise definition of the software functionality. Formal specification methods construct software in a stepwise fashion, adding more details in each step of refinement.

A component model defines standards for properties that individual components must satisfy and methods, and possibly mechanisms for composing components. This generic definition allows existence of different component models. The domain of embedded systems includes a large variety of systems (from ultra small computer-based devices to large systems monitoring and controlling complex processes.) and requirements; accordingly there is a need for development of different component models that provide support for providing solutions that meet these requirements.

2 Related work

2.1 Examples of existing generic component models

Nowadays there are many different component models. Some component models are aimed for specific application domains and some of them are for general purpose. However, there is third type of component models – generic component models, which can be used for instantiation of particular component models. CBD’s cluster aim is to develop a generic component model (ProModel) suitable for development of particular class of real-time embedded systems.

Here is a brief overview of two existing generic component models and their main characteristics:
Robocop is a component model developed by the consortium of the Robocop ITEA project, inspired by COM, CORBA and Koala component models. It aims at covering all the aspects of the component-based development process for the high-volume consumer device domain. Robocop component is a set of possibly related models and each model provides particular type of information about the component. The functional model describes the functionality of the component, whereas the extra-functional models include modelling of timeliness, reliability, safety, security, memory consumption, etc. Robocop components offer functionality through a set of ‘services’ and each service may define several interfaces. Interface definitions are specified in a Robocop Interface Definition Language (RIDL). The components can be composed of several models, and a composition of components is called an application. The Robocop component model is a major source of for ISO standard ISO/IEC 23004 for multimedia middleware.

The SOFA (Software Appliances) is component model developed at the Charles University in Prague. A SOFA component is specified by its frame and architecture. The frame can be viewed as component - a black box and it defines the provided and required interfaces and its properties. However a framework can also be an assembly of components, i.e a composite component. The architecture is defined a grey-box view of a component, as it describes the structure of a component until the first level of nesting in the component hierarchy. SOFA components and systems are specified by an ADL-like language Component Description Language (CDL). The resulting CDL is compiled by a SOFA CDL compiler to their implementation in a programming language C++ or JAVA. SOFA components can be composed by method calls through connectors. The SOFA 2.0 component model is an extension of the SOFA component model with several new services: dynamic reconfiguration, control interfaces and multiple communication styles between the components.

2.2 Functional property refinement and compositionality of functional property refinement

For the systematic stepwise development of components, the refinement relations are the key to formalize the development steps and the development process. In general, there are three refinement relations: property refinement, glass box (implementation) refinement and interaction refinement.

First of all, a mathematical notation i.e. formal specification of a component is needed. There is no commonly accepted and sufficiently precise definition of a component. One possible definition coming from Manfred Broy is “A component is an active information processing unit that communicates asynchronously with its environment through its interfaces syntactically characterized by a set of input and output channels”

Let I and O are accordingly the set of input and output channels of the component F, then a specification of a component should define:

- its syntactic interface,
• its behaviour by a specifying formula relating input and output channel valuations.

![Diagram]

Furthermore, a data type, indicating the type of messages that are sent through the channel, is associated with every channel in the set $I \cup O$. In addition, a stream is defined as a sequence of messages sent over a channel during the lifetime of a system. Since the communication of concrete system takes place in some time frame, timed streams are used.

The interface i.e. the black box behaviour of a component can be denoted as I/O- function that defines relation between the input streams and output streams of a component:

$$F : I \rightarrow O$$

The function yields a set of valuations for the output channels for each valuation of the input channels.

Property refinement allows replacing interface behaviour with one having additional properties. With every new step more and more sophisticated properties for components are added, until a desired, more restricted behaviour is specified.

Given interface

$$F : I \rightarrow O$$

is refined by a behaviour

$$\hat{F} : I \rightarrow O$$

if $\forall$ stream $x \in I : \hat{F}.x \subseteq F.x$

One possible general form of composition for components would be parallel composition with feedback. For given I/O functions with set of disjoint output channels

$$F_1 : I_1 \rightarrow O_1$$  \hspace{1cm} $$F_2 : I_2 \rightarrow O_2$$

parallel composition ($\otimes$) with feedback can be defined by I/O function

$$F_1 \otimes F_2 : I \rightarrow O$$

where $I = (I_1 \cup I_2) - (O_1 \cup O_2)$ and $O = (O_1 \cup O_2) - (I_1 \cup I_2)$.

As a consequence the rule for compositional of property refinement can be formulated as
\[
\hat{F}_1 \subseteq F_1 \quad \hat{F}_2 \subseteq F_2 \\
\hat{F}_1 \otimes \hat{F}_2 \subseteq F_1 \otimes F_2
\]

3 Central Literature and Seminal Papers

Since the field of formal methods is quite old, today there are many papers and books in this field. Furthermore, although CBSE field is a quite new field of research, there are also many resources of literature in this field. Here is a brief list of seminal papers and books in these fields:

Literature on CBSE:

- D.Illroy, 1968/9, Report from NATO Conference on SE
- Ivica Crnkovic, Magnus Larsson – Building reliable component-based software systems, Artech House, 2002

Literature on Formal Methods:

- Wing, J A Specifier's Introduction to Formal Methods. IEEE, September 1990
- Gries D. and Schneider F. A Logical Approach to Discrete Math. Springer 1993
4 Key Conferences and Leading Research Groups

4.1 Key conferences

In general, my research project is within the area of component-based software modelling and design for real-time embedded systems. The project is initially intended for embedded real-time system, however my research area is mostly connected with component based software engineering, formal methods and model-checking analysis for component models. Accordingly, conferences related to these fields are of main interest.

- **ICSE** (International Conference on Software Engineering) - premier software engineering conference, providing a forum for researchers and practitioners to present and discuss the most recent innovations, trends, experiences and concerns in the field of software engineering

- **ESEC/FSE** (European Software Engineering Conference and Foundations on Software Engineering Conference)

- **QoSA** (International conference on Quality of Software-Architectures) - topics include but are not limited to for instance: architecture design and implementation, component design and implementation, architecture evaluation and architecture management

- **International CBSE symposium** – has a track record of bringing together researchers and practitioners from a variety of disciplines to promote a better understanding of CBSE from a diversity of perspectives, and to engage in active discussion and debate.

- **Euromicro SEAA** (Software Engineering and Advanced Application) – general software engineering conference that addresses also CBSE

- **MODELS** (Model Driven Engineering Languages and Systems) – series of conferences devoted to the topic of model-driven engineering, covering both languages and systems used to create complex systems.

- **COMPSAC** (International Computer Software and Applications Conference) - focuses on highly dependable and secure software systems in order to address the foundations, methodologies, and mechanisms that support the design, modelling, and evaluation of software systems that are dependable and secure to the desired degree with acceptable performance.

- **FORMATS** (International Conference on Formal Modelling and Analysis of Timed Systems) - topics include but are not limited to for
instance: foundations and semantics of timed systems and languages, methods and tools for analyzing timed systems and application.

- **TACAS** (International Conference on Tools and Algorithms for the Construction and Analysis of Systems) - topics include but are not limited to for instance: programming languages, formal methods, hardware and software verification, static analysis and real time systems.

- **SEFM** (International conference on Software Engineering and Formal Methods) - The aim of this conference is to bring together practitioners and researchers to advance the state-of-the-art in formal methods, to scale up their application in software industry and to encourage their integration with practical engineering methods.

- **ICFEM** (International Conference on Formal Engineering Methods) - aims to bring together those interested in the application of formal engineering methods to computer systems.

### 4.2 Leading research groups and researchers

There are many research groups and researchers that are working in the field of component based development. Some of them are directly connected with real time systems.

- **TU/e Eindhoven Team** (Dieter Hammer, Egor Bondarev, Michel Chaudron) – research interests are: construction of embedded real-time systems, ROBOCOP project whose aim is to develop a component based software architecture for the middleware layer of high volume embedded appliances.
- **SOFA team** (Charles University, Prague, Czech Republic) – working on generic hierarchical component model SOFA 2.0
- **Palladio team** (Universität Karlsruhe (TH), Germany; leader Ralf Reussner) – working on Palladio component model (PCM), which is a metamodel for a domain specific language to describe systems build using components, and early design time predictions of Quality of service (QoS) attributes.
- **Fractal team** – working with Fractal component model which was designed and developed by France Telecom R&D and INRIA.
- **Astist2 NoE** real-time components cluster team and testing and verification cluster team. In general the main objective of Artist2 NoE is to set a scientific community in the area of embedded systems.
- **Rob Van Ommering** – working in general with component based architectures and architectural description languages, especially in the area of resource constrained systems. (ex. KOALA component model)
• Kung –Kiu Lau – in general working in the field of CBSD

Furthermore, here are some of the research groups and people who are working with formal analysis of systems, model checking and in general formal methods:

• **UPPAAL team** - Uppaal is an integrated tool environment for modelling, validation and verification of real-time systems modelled as networks of timed automata, extended with data types developed by the Department of Information Technology at Uppsala University, Sweden and the Department of Computer Science at Aalborg University in Denmark

• **Paula Inverardi** – working on application of formal techniques to the development of software systems, including software specification and verification of concurrent and distributed systems, deduction systems, and software architectures. Current research mainly concentrates in the field of software architectures specifically addressing the verification and analysis of software architecture properties, both behavioural and quantitative.

• **Rajeev Alur**, University of Pennsylvania, USA – working with model checking and refinement of real-time systems

• **Edmund Clarke** - Model Checking at Carnegie Mellon University

• **Thomas Henzinger** - EPFL, Switzerland and University of California at Berkeley, USA (Formal analysis of real-time systems)

5 Research issues

5.1 Detailed Description

The design process of embedded systems is often viewed as a sequence of steps that transforms a set of specifications described informally into a detailed specification that can be used for manufacturing. All the intermediate steps are characterized by a transformation from a more abstract description to a more detailed one. Unfortunately, the descriptions of the design at its various stages are often informal and not logically connected by a set of precise relationships. There is a broad range of potential formalizations of a design, but most tools and designers describe the behaviour of a design as a relation between a set of inputs and a set of outputs. This relation may be informal, even expressed in natural language and may result in unnecessary redesign. In my opinion, a formal model of a design should consist at least of the following two parts:

1. A functional specification, given as a set of explicit or implicit relations which involve inputs, outputs and possibly internal (state) information.

2. A set of properties that the design must satisfy, given as a set of relations over
inputs, outputs, and states, that can be checked against the functional specification.

The functional specification fully characterizes the operation of a system. A property is an abstraction of the behaviour along a particular axis. Identification of requirements is performed as in traditional development. However, in the component-based approach of real-time embedded system design, the mapping between system and component requirements is important. Requirements for components should be identified during system requirements elicitation, in order to reuse existing components. The development of an adequate technology for component-based development faces many challenges. This is in particular true for real-time embedded systems. Accordingly, the aim of CBD’s cluster project is to build a framework for component models suitable for the development of predictable real-time embedded systems.

Since the domain of embedded systems contains large variety of systems, which have many common characteristics but also differences, there is a need for developing a meta-model of a generic component model from which different component models can be instantiated. The meta-model is the declaration of the generic component model and in it we will formally define the basic architecture elements of the generic model. Today only few component models have specified metamodels. Usually, in these metamodels only the structure view and at most the binding of the components is covered (Picolo, Fractal, SOFA). As a result, these metamodels are very difficult to be analyzed. Therefore, part of my research, as well as Séverine, will be to study existing meta models of the component models and to provide a global meta model which will cover several views of the component models: structure, packaging, binding, deployment and execution.

Generic component model defines:
- generic component properties (functional and non-functional)
- generic interfaces these components have
- generic types of interaction (client-server, pipe and filter…)
Thus, a generic component model is a superset or an abstraction from a number of real component models and it encapsulates the principal characteristics of these models. This generic component model can cover several different domains. The idea is on this level I to describe several properties and behaviours. In order to do this, different languages should be used. From this generic component model different component models will be instantiated. Each component model is a subset of the generic component model and it includes:

- specific component properties
- specific interfaces the components have
- specific types of interaction

The properties for specific component models will be identified from the requirements coming from specific domains (automotive, telecommunication,…).

On the other hand, in my work, from a formal methods perspective, I will focus on two particular issues:

- Component and property specification: in the context of embedded systems, it is obvious that interface specifications of components must go beyond syntactic information and include functional and extra-functional characteristics and requirements. For real-time systems the temporal attributes of components and systems are of main interest. For embedded systems, the properties specifying the resources and the properties related to dependability (reliability) are important. However, there is still no consensus about how components for real-time systems should be specified. I intend to analyze particular component models either via model-checking techniques, or by refinement, which may subsume not only functional transformations of the system model, but also the addition of new constraints and properties to existing ones and the study of their impact on other system properties.

- Prediction of system properties from component properties: Even if we assume that we can specify all the relevant properties of components, it is not necessarily known how they will determine the corresponding properties of systems of which they are composed. This is especially true for properties like timing, reliability, maintainability etc. Moreover, existing component models do not provide support for predictable composition. In this, one should aim for interfaces providing full functional and extra-functional specifications of components. I will investigate ways in which and the extension to which various component properties are compositional.

The overall aim is by adding new properties and new constraints on the existing ones in the system to make it more predictable.
5.2 Research Method

The first step of my research is to study the existing component models and their metamodels. Based on the requirements coming from the industry sector (which is working with real-time embedded systems), a generic component model (ProModel) suitable for instantiation of different component models will be developed. On this level I will use different languages for describing different properties in order to formulate the semantics of the generic component model. Particular properties will be studied and mathematically specified. The next step, will be on the lower (component model) level to perform some property refinement (functional properties, reliability...) and to try to reflect the results from this level on the higher (generic component model) level.

5.3 Research Questions

- How to instantiate component models from the generic component model and how efficient this instantiation can be performed?
- How to specify different parameters of the components?
- How to model non-functional properties?
- How new properties reflect on the system?
- Is it possible to make the systems more predictable by performing property refinement? If we add some restrictions and some rules how the predictability of the system behaviour will be changed?

5.4 Hypothesis

- By performing property refinement or adding some properties more predictable systems can be built
- Meta-models describe several views of component models
- Composition of properties is different for different types of properties

5.5 Expected results

- Classification framework for component models
- Meta model - specification of the basic elements of the generic component model
- Semantics of the generic component model
• By using refinement techniques easily to define some component models which are suitable for specific usage

6 Activities

6.1 Courses

Taken:

• PhD School
• Progress Graduate Course

Ongoing:

• Research Planning
• Software Engineering Course
• Advanced Component Based Software Engineering Course
• Formal Languages, Automata and Models of Computation

Planned to Take:

• Research Methodology in Computer Science
• Real Time Systems, Advanced Course …

6.2 Planned Publications

The general idea is to write around 3 papers per year with different maturity level and to submit them to a workshop, conference or even better journal paper. The planned papers till Licentiate Defence are the following:

Paper #1: A Classification Framework for Component Models

Abstract. The essence of component-based software engineering is embodied in component models. Component models specify the properties of components and mechanism of component compositions. In a rapid growth, a plethora of different component models has been developed, using different technologies, having different aims, and using different principles. This has resulted in a number of models and technologies which have some similarities, but also principal differences, and in many cases unclear concepts. Component-based development has not succeeded in providing standard principles, as for example object-oriented development. In order to increase the understanding of the concepts, and to easier differentiate component models, this paper provides a Component Model
Classification Framework which identifies and quantifies basic principles of component models. Further the paper classifies a certain number of component models using this framework.

**Paper #2:** Extension of the paper “A Classification Framework for Component Models” possibly into a Journal Paper

**Paper #3:** Paper on metamodeling of component models with different views (structure, packaging, binding, deployment and execution)

**Paper #4:** Paper on language specification for describing components and their interaction in order to model embedded systems

**Paper #5,6:** Papers on defining semantics and adding new properties in the generic component model by using property refinement in order to easily instantiate and add new restrictions to component models

### 6.3 Coorporation

Since my research is done within Progress project and it is partitioned between the two interrelated research directions within the Component-based Development (CBD) Cluster: Component-based Software Engineering (CBSE) and Component Verification, a corporation is planned with academia and industry. Some of the universities which will be interesting and it is very possible to corporate with are:

- Universität Karlsruhe (TH), Germany
- Charles University, Prague, Czech Republic - Distributed Systems Research Group
- TU/e Eindhoven
- The University of Manchester - Kung–Kiu Lau
- Department of Computer Science, Uppsala University

Furthermore, the industry should not be excluded from the corporation circle. Accordingly a corporation with ABB, CC systems, Philips or even possibly Ericsson is of great interest.

### 6.4 Conference Trips

In general, conference trips are done when a paper is accepted on the corresponding conference. Usually, well-known and respected conferences or workshops are of primary interest. However, especially in the beginning of the PhD studies it is also recommended students to attend some conferences and workshops even if their paper is not accepted or if they have not submitted a paper for that conference. Accordingly some of the planned conference trips are:
7 Time Plan and Milestones

2006 October – December

- October – enrolled as a PhD student
- Courses: PhD School Course, Progress Graduate Course
- Progress trip to Santander, Spain
- International Research Training Groups Workshop 2006 on Dependability and Software Engineering in Dagstuhl, Germany

2007 January-July

- Paper#1 “A Classification Framework for Component Models”
- ARTIST2 – MOTIVES, Trento, Italy
- Paper#3 - metamodeling of component models with different views (design time, deployment time and binding). Review of component metamodels
- Courses: Software Engineering, Advanced Component Based Software Engineering, Research Planning and Formal Languages, Automata and Models of Computation
- June 25-28 Attend ITI Conference in Dubrovnik, Croatia

2007 August- December

- September 3-7 Attend ESEC/FSE Conference in Dubrovnik, Croatia
- Paper#4 – Generic Component Model
- Paper#5 – adding new properties by property refinement in order to make the component models more predictable
- Courses: Research Methodology and Real Time Systems - Advanced Course

2008

- Licentiate Proposal
- Take 10-15 credits of courses
- At least one publication
• Licentiate Defence

2009 (to be planned)

• Take 10 credits of courses
• At least one publication

2010 (to be planned)

• Take 10 credits of courses
• At least one publication
• PhD proposal

2011 (to be planned)

• PhD Defence
8 References


5. M.Broy; “Modular Hierarchies of Models for Embedded Systems“, Proc. of MEMOCODE’03, Mont Saint-Michel, France

6. M.Broy; “Compositional Refinement of Interactive Systems Modelled by Relations“, Proc. of COMPOS’97, Bad Malente, Germany

7. J.Liu, H.Miao; ”A Strategy for Component-based Modelling and Refinement“, Proc. of ICECCS'05, Shanghai, China


11. D.Illroy, 1968/9, Report from NATO Conference on SE

