



LARIS/EISTI Laboratory (Information and Systems Research Laboratory)

COMPUTER SCIENCE RESEARCH ACTIVITIES

Hubert KADIMA
Director of Research

1. MAIN RESEARCH AXIS

All researches carried out at our Computer Science Research Laboratory leans on the two following research axis:

- **Design and formal verification of component-based complex systems (distributed, mobile, real-time and hybrid systems) and application to mechatronics, automobile, aeronautics and telecommunications industrial problems.**
- **Semantic-based systems architectures and applications: Ontology-based systems architectures and application to e-learning systems, semantic integration of data and services, social networking ...**

The academic team is involved in national and european research programs related to these two main research axis.

1.1. DESIGN AND FORMAL VERIFICATION OF COMPONENT-BASED COMPLEX SYSTEMS

The main research themes of this axis are:

- **Modeling and component-based Design of complex systems.**
- **Formal verification of complex systems constraints and properties**
- **Integration of formal and semi-formal methods and Tools.**
- **Diagnostic and automatic generation of tests based on formal specifications**
- **Distributed Architectures: performances and security concerns in Peer-to-Peer, Grid Computing and Cloud Computing systems.**

1.1.1. Modeling and component-based design of complex systems

Component-based system modeling is being used more and more in designing of distributed and reactive complex systems. They are based on the usage of existing component (COTS), software and materials, patterns or frameworks which facilitate the design of complex systems and significantly contribute to the reduction of development costs.

The main problem system engineers face is the one of the confidence assembly of these components (interface problems, integration, compatibility etc...). We address models which allow property specification, both functional and non-functional, for system components to give to the developer a certain level of confidence in the reliability of the components assembly of hybrid complex-system designing process.

In the design process, the passage from specification to implementation happens in the framework of MDA architecture (*Model-Driven Architecture*) through consecutive refinements which allow us to go from PIM (*Platform Independent Model*) to PSM (*Platform Specific Model*) following different UML 2.0 profiles.

Keywords:

Component-based modeling, real-time formal system verification, Requirement Engineering and Traceability, Design Tools, Codesign, virtual prototyping/simulation, UML metamodel and profile. Modeling engineering, UML 2.0, SysML, Design patterns, Frameworks; MDA/MDE (Model-Driven Architecture/Model-Driven Engineering); Contractual programming/metamodeling.

1.1.2. Formal verification of complex systems constraints and properties

Formal verification is of crucial significance in the development of complex systems. As a result, the formal verification of hardware and software has become a promising area for research and industrial applications.

Model-based verification methods can be directly applied to systems to check formally their constraints and properties with appropriate languages and tools.

1.1.3. Integration of formal and semi-formal methods and tools

Despite their well-know advantages, formal methods are discarded by a large number of practitioners because of their notations and a lack of methodological support. To increase their industrial usage, a possible solution could be to integrate them with semi-formal methods and more particularity with *UML2/SysML*. During to design process of complex systems, different approaches present a promising blend of semi-formal and formal methods.

The necessity of tool support for formal methods is widely accepted. However, much existing tool support fails to exploit the advantages that formality brings. Our studies include all aspects of integration of different notations, paradigms, and tools, including integration of state-based and behavioral formalisms, and the formal strengthening of informal notations. We foresee the establishment of the integration of certain tools and formal methods used at this time (*language B*, *Simulink*, RTL tools, EDA electronic tools...) and UML2/SysML in preparation for *mechatronic* systems development. The incorporation of these tools will be in the Eclipse environment.

We are particularly interested in how integrating formal methods can facilitate tool support for the development process, either by integrating tools or exploiting the particular languages combined.

Key words:

Formal methods (abstraction, proofs, model-checking, transformations) ; formal real-time system verification ; formal validation techniques ; verification and validation processes and tools (pproofs, tests etc...) ; contractual/metamodeling programming; Language B ; temporal logic ; PBSE (Proof-Based System Engineering); real-time UML profiles; Time delayed algebraic processes.

1.1.4. Diagnostic and automatic tests generation from formal specifications.

We address the problem of diagnosing complex discrete-event systems. Given a flow of observations from the system, the goal is to explain those observations by identifying and localizing possible faults. Several model-based diagnosis approaches deal with this problem but they need the computation of a global model which is not feasible for complex systems. We are studying new approaches which permit to carry out on-line diagnosis without computing the global model.

Automatic test generation from formal specification would allow us to improve testing coverage of complex systems. The generation methods are based on approaches using formal specification techniques. The effectiveness of these different approaches is evaluated in the engineering validation process.

Key words:

Formal methods (abstraction, proof, model-checking, transformations) ; formal real-time system verification ; real-time modeling properties (components, systems) ; formal validation techniques ; verification and validation processes and tools (proofs, tests etc...) ; certification processes and tools; PBS (Proof-Bases System Engineering); Automatic test generation based on simulation specifications.

1.1.5. Distributed Architectures: performances and security

We address performances and security problems in designing P2P, Grid Computing and Cloud Computing systems.

Peer-to-Peer systems benefit from and share the resources owned by systems that are distributed around the Internet. Examples of such technologies include in general, any large-scale distributed system characterized by decentralization and sharing of resources.

Grid computing enables the sharing of distributed computing and data resources such as processing, networking and storage capacity to create a cohesive resource environment for executing distributed applications.

Cloud computing is a compelling way to take advantage of web based applications. Deploying modern web applications across a cloud infrastructure enables a new level of agility that is very difficult to accomplish with traditional silo computing model. New Computing models for virtualization and cloud require a very scalable, resilient and open network infrastructure.

Key words:

Peer-to-Peer Architecture; replication algorithms and optimist reconciliation ;P2P system security: SAML, WS –Security...; P2P network mobile access ; Grid Computing ; Cloud Computing architecture;

1.2. SEMANTIC-BASED SYSTEMS ARCHITECTURES AND APPLICATIONS

The main research themes of this axis are:

- **Ontology and semantic information research in web**
- **Modeling and semantic integration of data and services**
- **Social Networking**

1.2.1. Ontology and semantic information research in web

Ontology models a domain and provides vocabulary for needs specification of one (or many) target application. Ontology guides the development of operational systems. Methods and specific languages can be used to model and manipulate ontologies (RDFS, OWL...)

Work in this area focuses on:

- The representation of knowledge and reasoning
- Semantic indexation and usage of metadata in intelligent information research
- Models and tools in knowledge engineering for learning – *elearning*
- Architecture, norms and LMS standards (*Learning Management Systems*).

1.2.2. Modeling and semantic integration of data and services

Work in this area focuses on:

- Composition architecture: composition algebra, dynamic service oriented compositions, semantic coherence...
- Standards, modeling and semantic composition of business processes, ebXML/SCM (*Supply Chain Management*), Rosettanet
- Semantic integration of data and services (orchestration, mediation...)
- Semantic-based composition and execution of web services

Web Services have received much interest due to their potential in facilitating seamless business-to-business or enterprise application integration. Of particular interest is the Web Service Composition and Execution process - the creation of a workflow that realizes the functionality of a new service and its subsequent deployment and execution on a runtime environment. A significant number of solutions have been proposed in the literature for composition and execution of web services. New approaches based on semantic web technologies will be proposed in a *Cloud Computing* environment.

1.2.3. Social Networking

We are interested in topics related to semantic-based systems that have an impact on groups, organizations and social networks. Relevant issues include design, implementation, deployment, evaluation, methodologies, and effects of these systems.

All our current researches lean upon several national and european projects.