ERCIM “Alain Bensoussan” Fellowship Scientific Report

Fellow: Delphine Longuet
Visited Location : CWI, Amsterdam, The Netherlands
Duration of Visit: 9 months

I - Scientific activity

During my ERCIM fellowship at CWI in SEN3 group, I was involved in a work related to the CREDO european project. This project aims at developping theory and tools for the modelling, the implementation and the verification of dynamically reconfigurable component-based systems. My work deals in particular with verification and validation aspects.

In the framework of this project, components are modelled following an aspect-oriented approach in the Creol language. In this language, a component is a collection a objects, each equipped with its own processor. These objects communicate with each other in an asynchronous way, by method calls. It means that when a message (a method call) is received by an object, it is not directly executed but stored in a queue. This allows to simulate distributed applications in a realistic way. When received, method calls are stored in a "bag", without any particular scheduling strategy. The next method to be executed is then chosen randomly. In the Creol language, it is also possible to suspend the execution of a method by an object in order to release the processor when it is blocked waiting for an answer for instance, or waiting for a condition to hold. The execution of the method is then put back into the bag of calls while other methods can be executed. The execution of the pending method will be able to continue when the answer arrives or the condition holds.

For a given bag of calls, the order in which methods will be executed by the object is unknown a priori. In general, the method calls an object will receive and the order in which it will receive them is impossible to determine. To study the behaviour of such a system, in order to validate or to verify it, one may for instance consider all the possible scheduling for method executions. This needs to consider very large models, or to work at a high level of abstraction to express properties on the behaviour not depending on the order in which methods are executed. One may also decide to study a system according to a particular scheduling strategy. This allows to restrict the non-determinism inherent to each object.

Following this approach, a first study has been done about the schedulability of an object when it is added time constraints and deadlines for messages. An object can then be more abstractly modelled as a timed automata with deadlines. The environnement of an object, called a behavioral interface, is also specified by a timed automata. A scheduling strategy being chosen, it is possible to study the conditions under which an object is schedulable, i.e. executes all its tasks before their deadlines. To ensure the schedulability of an object according to a specific behavioral interface comes to check that a particular state of the system composed of the object and the interface is reachable, which is done using the Uppaal model-checking tool for timed systems.

Once each object is proved to be schedulable, this property must be proved on the whole system. This comes to prove that the actual environnement of each object in the system is compatible with its behavioral interface. We check compatibility using testing methods, which will not allow to prove

compatibility of the behavioral interfaces but to ensure a certain confidence degree in their design. The composition of behavioral interfaces can be considered as a specification of communications between objects of the system. Checking compatibility then comes to check that every behaviour of the system, described as a sequence of actions called a trace, is allowed by the behavioral interfaces. This is done by submitting to the system tests representing a given reachability property. Uppaal is used to generate a trace of the behavioral interfaces satisfying the property, from which a test case is built and then submitted to the system. We provide a test generation algorithm as well as proofs of correctness and non-laxness of tests cases.

II- Publication(s) during your fellowship

Mohammad Mahdi Jaghoori, Delphine Longuet and Frank S. de Boer
Schedulability and Compatibility of Real-Time Asynchronous Objects
Submitted to Real-Time Systems Symposium (RTSS'08).

Abstract. We apply automata theory to specifying behavioral interfaces of objects and show how to check schedulability and compatibility of real time asynchronous objects. The behavioral interfaces of real time objects specify (the order and timings of) the messages an object may send and receive. Each object is checked against its behavioral interface; first, to guarantee its correct output behavior, and second to make sure that every message it may receive is processed within the designated deadline (schedulability analysis). Next, we propose a new technique for testing whether every object is used as expected (i.e., according to its behavioral interface) when combined with other objects (compatibility check). Compatibility additionally implies schedulability in the context of the actual system. The analyses are automated using the Uppaal model checker. Our method makes it possible to put a finite bound on the message queue and still obtain schedulability results that are correct for any queue length.

Mohammad Mahdi Jaghoori, Delphine Longuet and Frank S. de Boer
Schedulability and Compatibility of Real-Time Asynchronous Objects
Extended version of the previous paper to be submitted to an international journal.

III -Attended Seminars, Workshops, and Conferences

Attendance:

- Meetings of the Amsterdam Coordination Group (SEN3 group seminar), every two weeks at CWI.
- Process Algebra Meetings (SEN2 group seminar), every week at CWI.

Talk:

- *Testing Dynamic Systems from Modal Specifications*, Meetings of the Amsterdam Coordination Group, January 29th 2008, CWI.