

Master's Thesis Proposal

Background:

This master thesis project is a part of the COMET project, between Linköping and Mälardalen University [2].

The goal of the COMET project is to enable development of a configurable real-time database for embedded systems, i.e., enable development of different database configurations for different embedded and real-time applications.

COMET has been developed using a novel design method called ACCORD [1], which combines software engineering techniques, such as component-based and aspect-oriented software development into real-time system software development. ACCORD enables design of real-time systems by first decomposing a system into a set of components, followed by the decomposition of the real-time system into a set of aspects. Components and aspects in the ACCORD (thus, also in the COMET) are designed and implemented using a real-time component model called RTCOM [1].

The initial COMET configuration (baseline COMET), implemented so far, corresponds only to the component-based decomposition of an embedded real-time database system. Aspect-oriented decomposition has been performed on a conceptual level, but has not been implemented yet.

The baseline COMET consists of the following components: user interface component, indexing component, memory handling component, and transaction manager component. This baseline configuration does not support concurrent transactions, and it is developed for non-concurrent real-time systems, where only one task can be active at one point in time.

The goal of the project:

The goal of the thesis is to extend a COMET component library and create an aspect library by making a concurrent COMET configuration. This implies implementation of adequate scheduling and concurrency control policies into the baseline COMET framework. Moreover, the goal is also to perform a study of impact of the ACCORD design as opposed to pure component-based design of the COMET system. Hence, two implementations of scheduling and concurrency control in COMET should be made: one according to ACCORD and RTCOM, i.e., using both components and aspects, and the other according to component-based software development, i.e., using components only. It is expected that the project results in 2 RTCOM components, namely scheduling and locking components, and 4 aspects, namely scheduling and concurrency control (1 for each policy chosen), and 4 non-RTCOM components, namely 2 scheduling (one for each scheduling policy chosen) and 2 concurrency control components (where locking and concurrency control are encapsulated within one component). As scheduling and concurrency control are strongly related, we expect strong dependencies between the two that need to be handled in the project.

Finally, the project should result in the answer to the following question:

- ❖ What is the trade-off between using mechanisms within RTCOM together with aspects as opposed to usage of pure components (drawback and benefits in terms of maintainability, code size, memory footprint, exchangeability of features in the database)

Following steps/stages should be performed in the project:

(i) Background preparation stage:

- ❖ Study existing scheduling and concurrency control policies;
- ❖ Study ACCORD design method and get a full understanding of the RTCOM;
- ❖ Study current COMET implementation and design documentation;
- ❖ Make a detail time-plan with all steps of the project;

(ii) Design stage and refinement stage:

- ❖ Pick (at least) two scheduling and two concurrency control algorithms, with distinct characteristics (e.g., optimistic vs. pessimistic concurrency control policy). The choice should be motivated in terms how it fits the COMET, and ACCORD.
- ❖ Make a design of the concurrency control aspect, and the locking component. The locking component should, as defined by RTCOM, consist all mechanisms needed by both policies.
- ❖ Determine the crosscutting impact of the aspect to the existing COMET, i.e., to which degree the other existing COMET components should be modified to support concurrency control.
- ❖ Make a design of the scheduling aspect, and the scheduling component (the scheduling component should, as defined by RTCOM, consist all mechanisms needed by both policies)
- ❖ Determine the crosscutting impact of the scheduling aspect to the existing COMET, i.e., to which degree the other components should be modified to support concurrency control. – Emphasis is on the relation between concurrency control and locking component and scheduling component and scheduling aspect.
- ❖ Make a design of the concurrency control component (performing locking + concurrency control), and study or predict the impact of having concurrency control and locking into one component (as opposed to aspectual implementation) of the remaining COMET configuration.
- ❖ Make initial predictions of possible benefits and drawbacks of RTCOM approach as opposed to pure component-based approach.
- ❖ According the findings, at the end of this stage the thesis goals could be redefined to reflect the conclusions obtained while making the design decisions and observing the impact of the design to the system and the initial thesis goals.

(iii) *Implementation phase:*

- ❖ Implement both aspectual and component-based approach and (i.e., 2 concurrency control aspects and 2 scheduling aspects + locking component + scheduling component + concurrency control component)

(iv) *Performance analysis phase:*

- ❖ Develop adequate tests to answer the following question:
 - What is the trade-off between using mechanisms and RTCOM together with aspects as opposed to usage of pure components (drawback and benefits in terms of maintainability, code size, memory footprint, exchangeability of features in the database)
- ❖ Perform tests and derive conclusions

Background knowledge that would help candidate student:

- ❖ Basics in database systems
- ❖ Basics in real-time systems
- ❖ C and C++

Project set-up:

- ❖ Two students are needed to carry out the work, and they are, most likely, going to closely work on the project. One student should work on scheduling and the other on concurrency control issues.
- ❖ After the (i) phase of the project student should prepare a description of its thesis goals and the problem description
- ❖ After the (ii) phase of the project, the goals of the project are refined and could be adjusted further to reflect the workload and conclusions reached in (ii)
- ❖ During each phase of the project student should document her/his findings, conclusions, design, implementations, and results, so that the thesis writing process is initiated and carried out during each stage in the project

Proposed time-framework:

(i) *Background preparation stage* - 3 weeks

(ii) *Design stage and refinement stage* – 5 weeks

(iii) *Implementation phase* – 5 weeks

(iv) *Performance analysis phase* – 5 weeks

(v) *Thesis wrap-up* – 2 weeks

The project lasts for 20 weeks, and is done by both students in parallel.

Research areas covered by the project:

Work on the thesis provides a student with a deeper knowledge and understanding of the following areas of computer science/engineering:

- ❖ Advanced software engineering techniques, namely component-based software development and aspect-oriented software development
- ❖ Real-time systems
- ❖ Embedded systems
- ❖ Database systems

References:

- [1] A. Tesanovic, D. Nyström, J. Hansson, and C. Nörstrom, Towards Aspectual Component-Based Real-Time Systems Development, RTCSA 2003, Taiwan.
- [2] Project web-site:
http://www.ida.liu.se/labs/rtslab/projects/ARTES_EmbeddedDatabases/welcome.html