Real-time software for versatility, scalability and reconfigurability in complex embedded feedback control systems - MARCH

First year status report 2000-06-12 of project A4-9805

(The project was started on 990701, so actually a full year has not yet passed)

Research results in relation to research plan

General

The goal of the MARCH[1] project is to progress state of the art of software architecture for complex feedback control systems. A specific goal of the project is to provide a system architecture suitable for scalable, maintainable and reconfigurable mechatronic control systems. The research findings are in the end planned to be evaluated and partly verified on a real robotic system, namely the 4-legged robot Warp1, which is now made operational within CAS (Centre for Autonomous Systems).

The Warp1 robot, which is aimed at autonomous operation in difficult outdoor terrain, constitute a controlled system of significant complexity. The robot has 12 actuated joints, 5 on-board computers and a remote host running on a PC-machine. The distributed computer control system is based on CAN-communication. More than 30 sensors are used for estimating the internal state of the robot. Modularity is an important requirement on the whole system.

An important problem in developing mechatronic systems is the increasing complexity both in the requirements and in the design solutions. In generic terms, complexity can be defined by (1) the number of elements (2) the relationships between the elements, and (3) the degree of organization of the elements. In mechatronic systems, there is typically a large number of system constituent units from different engineering domains. All these units also have multiple interrelations resulting from the interrelated functional and non-functional quality attributes, the adopted functional design, the implementation scheme, and the target platform/infrastructure. From a system point of view, unsuccessful management of the complexity results in functionality and dependability related problems in both development and maintenance. Thus, the goal of the project in a concrete sense is to progress state of the art of software architecture by introducing mechatronics specific semantics and syntax.

Research activities

The research plan (for the first year) according to the original project proposal includes the following research tasks which are here complemented with corresponding comments about status and achieved results.

- Establishment of a requirement specification for the architecture. This task is performed strongly liked to analysis of the control requirements (as formulated at the corresponding point in time) as well as non-functional platform requirements. Deliverable: Specification.
  Status: Ongoing
  Results: Utilizing the results from work on generic software architecture to improve mechatronics software should be carried out with great caution due to the semantic gap between these two software application domains. For example, in generic software architecture, the real-time issues raised by feedback control applications have not been tackled. Hence, in the current phase of the research, the focus is on identifying the fundamental

1. Mechatronic ARCHitecture - MARCH - is the short form project name.

semantics and semantic interrelations of the typical constituent units in a mechatronics feedback control system and on establishing the fundamental structural interrelations of the units (i.e., functional, behavioural, temporal, spatial interrelations). Guided by the semantic information, the concept and theories from generic software architecture can be adapted to suit the feedback control software systems.

Exploring the semantic information has been well incorporated in the object-oriented software design paradigm. In this design paradigm, the semantic information is managed in terms of classes and their interrelations, such as inheritance, association and aggregation. The semantic contents of an OO software element include its attributes and operations (e.g., methods). The success of the OO paradigm has exemplified the importance of semantic exploration and management for both design and maintenance. Compared to the OO paradigm, the semantic of mechatronic feedback control software should be defined in a wider scope. This has been exemplified by the insufficiency of the OO methods in the design, upgrade and modification of large mechatronics software systems.

The aim is to establish an architecture design framework for mechatronics systems. The results of the current research will appear in the report “An architecture design framework for mechatronics systems”, which is one section of a Licentitate thesis planned for the fall 2000. The more specific aim of establishing an architecture for the WARP robot is somewhat delayed due to late hiring of the corresponding PhD-student within CAS. This work is thus currently ongoing.

- A state of the art study on software architectures for real-time feedback control systems. This study includes definition of concepts and notation (e.g. what is an architecture, which are the problems?, etc). Deliverable: Report.

  Status: Finished

  Results: An extensive state of art study has been performed and reported in [2] [1] [3]. The research concludes that the overall system structure in terms of system architecture is extremely important in handling the complexity of generic software systems. The power of architecture comes from abstraction, which sets the system information in a multi-viewed and hierarchical/top-down manner and supports comprehension, consistency analysis, decision-making and communication. A concluding hypothesis for further research is that the complexity can be managed by systematically using abstractions in terms of architecture. This would enable identifying system constituent units, revealing their interrelations and organizing them in a requirement consistent manner.

  In generic software design, architecture has been used to reveal the functional and behavioural interrelations of the software units, and to form the basis for modelling and establishing a component-based system development. Software architecture has also been used to involve all system stakeholders in the development and to achieve evolutionary/incremental system development. Last but not least, system integration and verification has been greatly facilitated with a properly developed software architecture. To support the design, some fundamental architectural principles (such as encapsulation/ modularization/ information-hiding) and the corresponding evaluation criteria (such as coupling and cohesion) have been widely accepted in generic software architecture design. There are also various architecture styles used to define mature solutions in managing/structuring the system constituent units and their interrelations, (e.g., black board, virtual machine, layer, client-server, pipe, etc.). To further support software architecture design, a set of ADLs (Architecture Description Language, such as MetaH, WRIGHT) and architecture design methods (such as SAAM and ATAM) have been proposed.

- Evaluation (without any implementation or at least only minor implemetation work) of existing relevant software architecture concepts with respect to the specification. This task

includes the establishment of an evaluation procedure. Here the AIDA modelling framework is expected to be very useful. Deliverable: Evaluation report.
Status: Ongoing
Results: The proposed architecture design framework for mechatronics systems will be evaluated with some well-known architecture solutions, including Simplex, TTA, and OSACA. This work is ongoing and initial evaluation results are to appear in the planned Licentitate thesis during the fall 2000.

Publications

[1] D.J. Chen, M. Törngren. *System Architecture in a Mechatronics Perspective*, Proc of SNART'99, Linköping, August 24-25, 1999.

[2] De-jiu Chen, System Architecture for Mechatronics Systems - A survey of the concepts, theories and methods with the focus on software, Technical Report, TRITA-MMK 1999:30, ISSN 1400-1179, ISRN KTH/MMK/R-99/30-SE.

Other reports

[3] D. J. Chen, Towards A Software Component Framework for Mechatronics Applications, The Component-Based Software Engineering – State of the Art, Graduate Course Report, Mälardalen University, Västerås 2000.

[4] D. J. Chen, Towards A Model-Based Evolutionary Design for Complex Mechatronics Systems, Konstruktionsmetodik, Graduated Course Report, KTH – LiTH, 2000.

Industrial/collaborative work

Due to changes in the experimental equipment (the Warp1 robot) being used in the project, there is no longer any interaction with the original industrial partner, Arcticus Systems AB. The reason being that the RUBUS real-time operating system is for various reasons not used in the current Warp1 robot.

The research topics of the project have been discussed with other companies such as SAAB Combitech Systems and SAAB Automobile. It is probable that closer interaction will be formulated with some or both of these companies in a later stage of the project.

The major element of collaborative work in MARCH is conducted together with CAS. Here, a new PhD student (Mats Gudmundsson) was hired in January 2000 according to the equal share basis between ARTES and CAS as outlined in the original project proposal. Also, there are additional research students within CAS studying archtitectural issues for mobile robots (wheel based systems). In recurrent discussions with the scientific advisory board of CAS, including industrial representation, the research issues of *systems integration* in robotic systems have been highlighted as very important. Software architecture as approached in MARCH is certainly an important basis for such systems integration.

A seminar on *Architectures for embedded systems* with a mix of industrial, international and university speakers was held at KTH 990922 as a joint arrangement between MARCH and the NUTEK funded DOORS project (finished by 990630). The seminar attracted some 65 participants of which a majority from swedish industry.

A doctoral course has been developed and is being carried out during the year 2000 at the Mechatronics lab. The course topic is the modelling of embedded real-time systems through UML, evaluating its capabilities. This course i partly based on seminars given by industrial partners.