has faithfully captured the (informal) system requirements. Diversity can help at this stage as well, by considering two independent formal specifications. A second form of diversity can thus be introduced at the level of specification; this will impact the whole development process.

The application of this idea to the railway signalling domain has provided a direct way of conceiving diverse specifications: the relay schemas that still constitute a common language for railway signalling engineers have been used for one version, while a more 'modern' and increasingly popular notation – UML sequence diagrams – have been used for the other. From these two specifications, two independent chains of verification/code generation/compilation/deployment have been implemented (Figure 2). The final comparison is made by running the set of official acceptance tests for the equipment developed on both versions.

The results of our experiments on the introduction of diversity in compilation have been encouraging, and their relatively low cost should facilitate industrial acceptance of the approach. In fact, the added cost of the first approach is limited to repeat compilation and testing on a Windows-based machine, with practically no cost for additional hardware or software resources. Moreover, replication of compilation and testing can be automated to a high extent.

In contrast, the introduction of diversity in specifications requires at least the additional effort of writing an independent specification. The overall cost of producing diverse specifications is therefore twice that of a single formal specification process. However, the higher costs of this form of diversity can be justified by lower testing and debugging costs due to the early discovery of design faults, and by the higher level of safety achieved.

Link:
http://fmt.isti.cnr.it

Please contact:
Stefania Gnesi
ISTI-CNR, Italy
E-mail: stefania.gnesi@isti.cnr.it

---

**Evaluation of Natural Language Requirements in the MODCONTROL Project**

by Antonio Bucchiarone, Stefania Gnesi, Gianluca Trentanni and Alessandro Fantechi

*We describe QuARS (Quality Analyzer for Requirement Specifications) Express, a customized version of the QuARS tool. It is designed to evaluate natural language requirements, can handle complex and structured data formats containing metadata, and is able to produce an analysis report with categorized information.*

On behalf of ERCIM, the FMT Group of ISTI-CNR has participated in MODCONTROL, a subproject of the recently concluded European Integrated Project MODTRAIN. MODTRAIN stands for Innovative Modular Vehicle Concepts for an Integrated European Railway System, and was the first Integrated Project to focus on joint European railway research. The objective of MODTRAIN was to define the functional, electrical and mechanical interfaces and validation procedures for a range of interchangeable modules that will form the basis for the next generation of intercity trains and universal locomotives. MODCONTROL addressed the standardization of an innovative Train Control and Monitoring System (TCMS) designed for future interoperable European trains.

Achieving high quality in the definition of software requirements is a must in the development of reliable and dependable software products. The availability of techniques and tools for the analysis of requirement documents may help to remove inconsistencies and ambiguities at as early a stage as possible. An automatic analysis of the requirements expressed in natural language should help to guarantee the successful outcome of a project by highlighting potential sources of ambiguity and other weaknesses.

In the context of MODCONTROL, we have developed the QuARS Express tool, a customized version of the QuARS tool (see ERCIM News No. 58), able to handle complex and structured data formats containing metadata and to produce an analysis report with categorized information. The new reporting feature improves on the simple text-based report provided by QuARS by exploiting HTML technology to produce structured hypertext pages.

Using QuARS Express, we analysed the functional and system requirements of TCMS, which included more than 5700 requirements. The results showed that an analysis based on QuARS Express not only identifies linguistic defects, but can also provide some indication of the diverse styles used by different partners to express requirements.
Evaluation of Natural Language Requirements in the MODCONTROL Project

by Antonio Bucchiarone, Stefania Gnesi, Gianluca Trentanni and Alessandro Fantechi

We describe QuARS (Quality Analyzer for Requirement Specifications) Express, a customized version of the QuARS tool. It is designed to evaluate natural language requirements, can handle complex and structured data formats containing metadata, and is able to produce an analysis report with categorized information.

On behalf of ERCIM, the FMT Group of ISTI-CNR has participated in MODCONTROL, a subproject of the recently concluded European Integrated Project MODTRAIN. MODTRAIN stands for Innovative Modular Vehicle Concepts for an Integrated European Railway System, and was the first Integrated Project to focus on joint European railway research. The objective of MODTRAIN was to define the functional, electrical and mechanical interfaces and validation procedures for a range of interchangeable modules that will form the basis for the next generation of intercity trains and universal locomotives. MODCONTROL addressed the standardization of an innovative Train Control and Monitoring System (TCMS) designed for future interoperable European trains.

Achieving high quality in the definition of software requirements is a must in the development of reliable and dependable software products. The availability of techniques and tools for the analysis of requirement documents may help to remove inconsistencies and ambiguities at as early a stage as possible. An automatic analysis of the requirements expressed in natural language should help to guarantee the successful outcome of a project by highlighting potential sources of ambiguity and other weaknesses.

In the context of MODCONTROL, we have developed the QuARS Express tool, a customized version of the QuARS tool (see ERCIM News No. 58), able to handle complex and structured data formats containing metadata and to produce an analysis report with categorized information. The new reporting feature improves on the simple text-based report provided by QuARS by exploiting HTML technology to produce structured hypertext pages.

Using QuARS Express, we analysed the functional and system requirements of TCMS, which included more than 5700 requirements. The results showed that an analysis based on QuARS Express not only identifies linguistic defects, but can also provide some indication of the diverse styles used by different partners to express requirements.
in natural language, thus laying the basis for improvements in the formulation of requirements.

The overall quality analysis process adopted in the project is shown in Figure 2 and is summarized in the following points:

- The project partners create a new project using IBM RequisitePro and insert the requirements with all the required attributes (Name, Text, Responsibility, Package etc).
- The different requirements are stored in a requirements file, one for each requirement class, i.e., Functional Requirements (i.e., FREQ) and System Requirements (i.e., SREQ).
- The IBM tool SoDA is used to generate a text document containing the requirements and the relevant attributes, which is saved in text format. A specific template has been defined for SoDA in order to allow QuARS Express to properly interpret the information contained in the generated document.
- The text file obtained is input to QuARS Express, which analyses the sentences (requirements) and gives as output the Defective Requirement Reports (DRR) for both FREQ and SREQ documents, together with the calculation of relevant metrics.

• If QuARS Express should point to some (possible) defect, the DRR should be filtered by experts in a ‘false defect survey’, in order to establish whether or not a refinement is really necessary. In this case, a new cycle of quality analysis may be initiated.
• Otherwise, the approved requirements document is released.

Links:
http://www.modtrain.com
http://quars.isti.cnr.it/

Please contact:
Gianluca Trentanni
ISTI-CNR, Italy
E-mail: gianluca.trentanni@isti.cnr.it

Development of Safety Software for the Paks Nuclear Power Plant

by Tamás Bartha and István Varga

Software development in a safety-critical environment requires that a rigorous process be followed in all phases of the system life cycle. Members of the Systems and Control Laboratory of SZTAKI are working together with experts at the Paks nuclear power plant in Hungary to develop new tools and support systems that will improve the dependability of the safety software.

The Systems and Control Laboratory (SCL) of SZTAKI has a long-term successful research and development collaboration with the Paks nuclear power plant to provide new methods and tools for the development of safety software in the plant. Some notable recent results of this work (listed according to their role in the development life cycle) are the following:


2. Development and implementation:
   - Microcontroller level: the hardware and software for the microcontroller-based smart test plugs of the Universal Test System (UTS)
   - Programmable Logic Controller level: the distributed control hardware and software of the new Primary Pressure Controller
   - Application level: the UTS test management software.

3. Testing:
   the Universal Test System.

Formal Verification of Functional Block-based Specifications
Each reactor unit of the nuclear power plant is supervised by a Reactor Protection System (RPS), which continuously monitors the nuclear process in order to intervene and safely shut down the unit in an emergency situation. The plant experts specify the safety functions of the RPS using the Functional Block Diagram (FBD) description method. The software for the RPS is then created automatically by a certified code generation process.
in natural language, thus laying the basis for improvements in the formulation of requirements.

The overall quality analysis process adopted in the project is shown in Figure 2 and is summarized in the following points:

• The project partners create a new project using IBM RequisitePro and insert the requirements with all the required attributes (Name, Text, Responsibility, Package etc).

• The different requirements are stored in a requirements file, one for each requirement class, i.e. Functional Requirements (i.e., FREQ) and System Requirements (i.e., SREQ).

• The IBM tool SoDA is used to generate a text document containing the requirements and the relevant attributes, which is saved in text format. A specific template has been defined for SoDA in order to allow QuARS Express to properly interpret the information contained in the generated document.

• The text file obtained is input to QuARS Express, which analyses the sentences (requirements) and gives as output the Defective Requirement Reports (DRR) for both FREQ and SREQ documents, together with the calculation of relevant metrics.

• If QuARS Express should point to some (possible) defect, the DRR should be filtered by experts in a ‘false defect survey’, in order to establish whether or not a refinement is really necessary. In this case, a new cycle of quality analysis may be initiated.

• Otherwise, the approved requirements document is released.

Links:
http://www.modtrain.com
http://quars.isti.cnr.it/

Please contact:
Gianluca Trentanni
ISTI-CNR, Italy
E-mail: gianluca.trentanni@isti.cnr.it

Figure 2: MODCONTROL evaluation process.

Development of Safety Software for the Paks Nuclear Power Plant

by Tamás Bartha and István Varga

Software development in a safety-critical environment requires that a rigorous process be followed in all phases of the system life cycle. Members of the Systems and Control Laboratory of SZTAKI are working together with experts at the Paks nuclear power plant in Hungary to develop new tools and support systems that will improve the dependability of the safety software.

The Systems and Control Laboratory (SCL) of SZTAKI has a long-term successful research and development collaboration with the Paks nuclear power plant to provide new methods and tools for the development of safety software in the plant. Some notable recent results of this work (listed according to their role in the development life cycle) are the following:


2. Development and implementation:
   - Microcontroller level: the hardware and software for the microcontroller-based smart test plugs of the Universal Test System (UTS).
   - Programmable Logic Controller level: the distributed control hardware and software of the new Primary Pressure Controller.
   - Application level: the UTS test management software.


Formal Verification of Functional Block-based Specifications

Each reactor unit of the nuclear power plant is supervised by a Reactor Protection System (RPS), which continuously monitors the nuclear process in order to intervene and safely shut down the unit in an emergency situation. The plant experts specify the safety functions of the RPS using the Functional Block Diagram (FBD) description method. The software for the RPS is then created automatically by a certified code generation process.