



## INNOVATION REPORT

# Market enabler for retargetable COTS components in the embedded domain

.....

***A general trend in the real-time and embedded (RTE) domain is towards complexity. Large, highly integrated and functionally extensive systems are developed with a constant need for reductions in costs and time to market. This tendency has established a need for overhauling development methods in the RTE area.***

Using commercial off-the-shelf (COTS) software components could be an answer to this concern. Effectively, COTS components are cheaper as they are reused several times. They are also safer, since they have been applied and validated extensively by different users.

However, several major obstacles prevent COTS software use from spreading in RTE systems development. First, the heterogeneity of components prevents seamless integration. Then, the poorness of components characterisation makes it difficult to choose the right component. Last, but not least, the absence of COTS use support tools and methodological guidelines discourage many developers.

The ITEA project MERCED aims to provide solutions to these concerns and, in this way, provide an impulse for the emergence of a European COTS software market.

### **Fundamental building blocks**

Real-time and embedded systems are becoming increasingly complex, with industry recognition of the need to have a transition from the design of individual devices to the design of overall systems resulting from the integration of reusable retargeted components. In a COTS-based development scheme, instead of developing software, designers browse catalogues of software components from multiple vendors and build complex systems by assembling COTS components.

Main benefits are development time and cost decreases. Reuse of software components increases productivity and quality; and components are more robust since they have been used by multiple users in different domains.

COTS software thus seems to offer major advantages to developers. However, several obstacles have to be overcome before COTS usage could spread in the RTE systems development field.

Firstly, to make possible a successful market for retargetable COTS components, it is crucial to understand the needs of engineers who will – or will not – identify, select, qualify or integrate COTS components. Such needs are not yet expressed and, moreover, no market research has yet studied such requirements. Therefore, it is quite difficult to come up with an adequate offer to meet the classes of needs that will be encountered in the embedded COTS market. Reuse not only means using an existing component – asset, COTS or other – but also implies the effective application of an existing asset within a system with, as a consequence, a reduction in the effort or cost of the solution provided. If this is not so, reuse is not a solution but a problem. A global infrastructure is thus necessary to support software component reuse among different systems and environments.



## INNOVATION REPORT

Then, RTE systems are required to support increasing functionality and are now often supposed to be part of a more general system and well integrated in it – e.g. a radar versus a surveillance system. The software architectures currently used to design such systems were primarily meant to take into account performance issues and are not structured sufficiently to support this increase in complexity. In addition, the lower parts – digital signal processing (DSP) implemented – are also becoming more and more complex, and more integrated with the upper part implemented on general purpose processors (GPP). The challenge is thus to move from specialised performance-centred approaches to a seamlessly integrated complexity-centred approach, while not losing adequate support for performance and time management in the move.

MERCED aims to answer all these concerns. It started in mid 2004 to gather significant industrial and academic actors in the European RTE systems area. MERCED partners aim to participate in the emergence of a European COTS software market by providing its fundamental building blocks.

Firstly, to ensure and validate the development of embedded systems based on RTE components, MERCED is targeting definition and implementation of an RTE component reuse environment prototype and a component-populated repository including both domain-specific and cross-domain components. This environment will be accompanied by a proper component reuse conceptual and methodological framework.

The second main part of MERCED deals with component implementation and execution issues. The aim is to design and prototype a specialised retargetable execution platform for RTE that could handle all non-functional aspects, thus enabling a separation of concerns between domain-specific processing (components) and non-functional logic (execution platform).

### The MERCED Components Reuse Environment for Real Time and Embedded

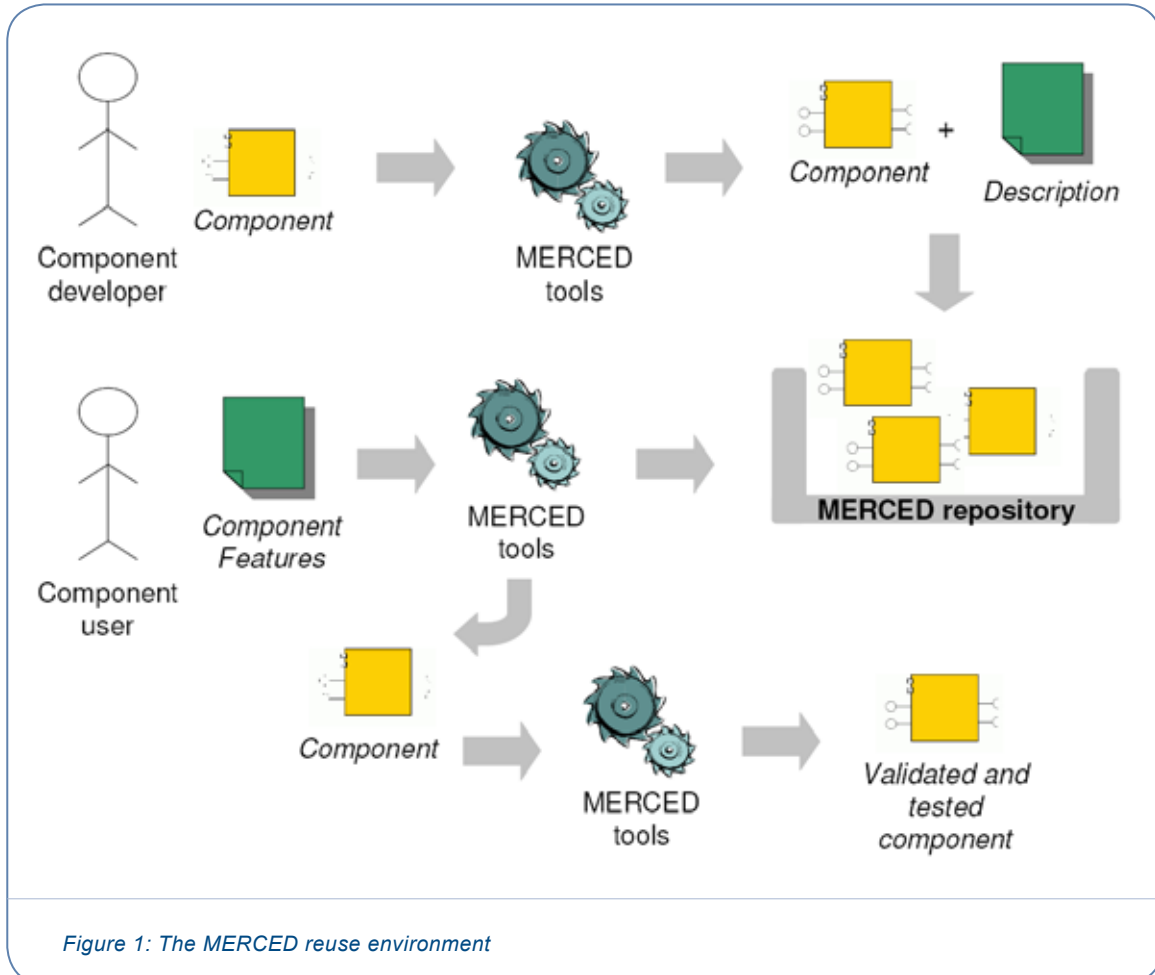
The overall goal of the MERCED components reuse environment is to provide a support for components storage and classification on the one hand, and components selection and retrieval on the other. In the proposed scheme, two actors are distinguished (*Figure 1*):

1. The **components developer** feeding the components repository with products. The developer has to provide a description with each component; this description has to be compliant with the MERCED RTE components taxonomy; and
2. The **components user**, who needs to retrieve specific components. The MERCED environment offers tools – such as a search engine – to support the user in his search, as well as in the steps that follow: component validation, simulation, and testing.

#### ■ Components classification and retrieval

MERCED will provide a component meta-model giving a common and unified component specification. This to ensure that all the information included within the infrastructure regarding components is provided in a unified manner. This meta-model will help automate many activities related to searching, retrieving, qualifying, testing and integration since which specification information can be manipulated is already known prior to use. The meta-model will be used to characterise components within the reuse environment. In this way, the information required for selecting and using components will be available to search tools. The meta-model will include extendable classification for testing concepts – mock-up environment, test-suite definition, test-driver components, stub components, qualification level and so on. In addition, a suitable approach

## INNOVATION REPORT



for component documentation and searching with extensibility capability will also be provided. The initial categorisation for non-functional and functional properties will be defined in such a way that it can easily evolve over time: proper classification extensibility mechanisms are thus to be defined and provided.

Once components are chosen and retrieved from the repository, they will be evaluated by an extensive tool chain providing validation and simulation facilities.

### ■ Component validation and qualification tools

To make the embedded real-time software development teams confident about reusing components, these elements must meet the qualification level needed. This qualification level is project-dependent and can be one of the component characteristics. For example, an avionics integrator might require a component certified at the 'DO-178B level A' level to be integrated into a critical subsystem. This qualification level can be part of the component description. Moreover, in the component selection process, the potential reuser has to ensure that the selected component description matches the needs, and will then also need to be confident that the component conforms to its description.

To achieve component qualification, the component supplier will need a test bench for the components being developed. This test bench will allow development of the qualification test suites needed. As the component is retargetable, the test suite also needs to be retargetable. In MERCED, a test platform will be provided at the hardware and operating-system (OS) abstraction

## INNOVATION REPORT

level. This platform will allow definition and possibly running of the component test suites in a virtual environment simulation. When the component is (re)targeted, a (re)targeted version of the tests on the real target can be run to assess explicitly the qualification level of the component for this new target.

### ■ System simulation tools

The selection of an existing component to be integrated in the designed system relies on a 'search, select and try' process. The 'search' and 'select' steps are achieved by a description of each existing component in the repository and associated searching and filtering tools based on this description.

The 'try' step consists of quickly verifying the component interacts as expected inside the system. Powerful means for the inspection of component interactions inside the system are needed and can be achieved in a simulation environment providing traces, break conditions, system-state inspection and other debugging features that will not be available in the final target or induce heavy supplementary overhead. The main features of the simulation environment need to allow logging and/or inspection of messages forwarded by the container to every component in the simulation. With a first try of the component in such an environment, the integrator will then be able to know easily if the component suits as it stands, needs a little tweaking to match perfectly or is definitely not usable in the present context, and why.

### The MERCED execution platform for Real Time and Embedded

The global purpose is to develop an execution framework for RTE systems. The base technology chosen for this framework is the common object request broker architecture (CORBA) component model [1], in its lightweight version [2]. This latter enforces an original execution model, based on a strict separation of concerns between business and non-functional aspects. All business logic is gathered in application components, while all non-functional processing is embedded in containers. Containers provide an execution context for components: they mediate all interactions to/from the external environment and, in doing so, shield components from any execution platform-dependent issues (Figure 2).

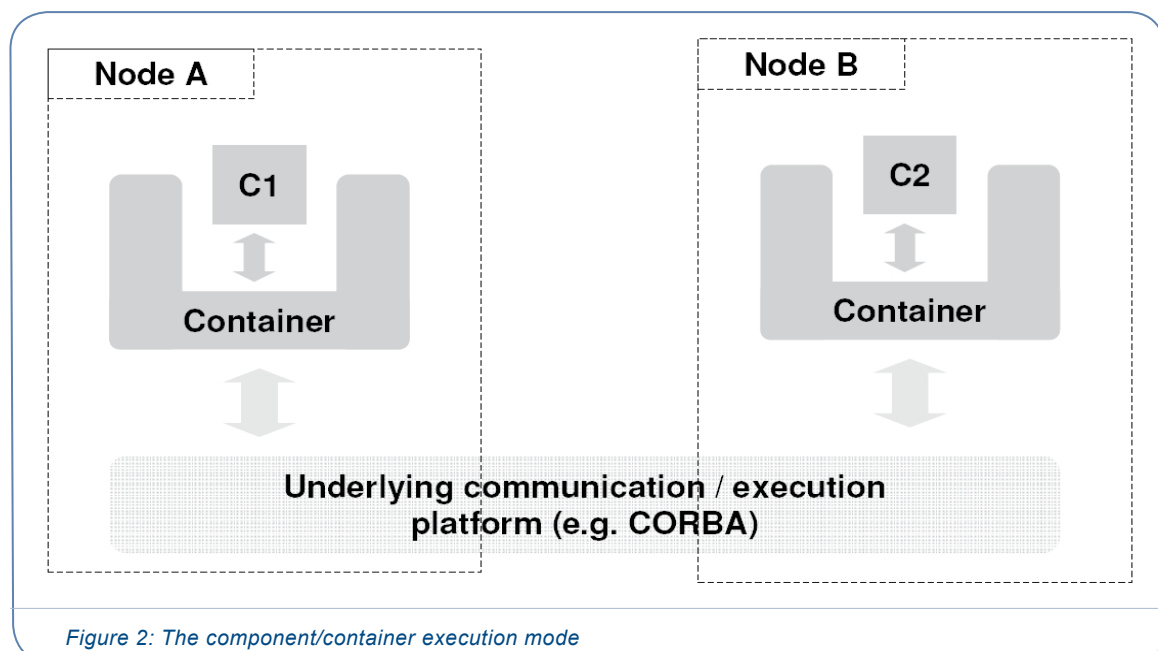
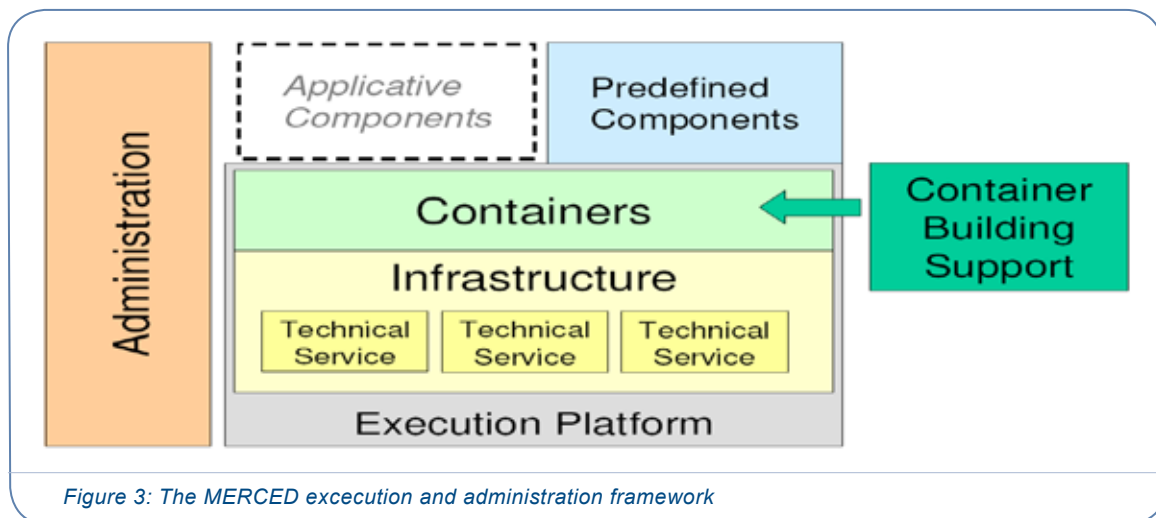


Figure 2: The component/container execution mode

## INNOVATION REPORT

The MERCED framework itself will be mainly made of the following parts (*Figure 3*):

- The execution infrastructure, comprising technical services;
- The containers to fill the gap between the components and the execution infrastructure;
- The administration service allowing assembly, configuration and deployment of the components and their containers;
- A tool allowing construction of the containers according to the application requirements; and
- Some predefined components, providing generic services.



*Figure 3: The MERCED execution and administration framework*

The main purpose of this part of the project will be to realise a reference implementation of this framework on top of real-time CORBA [3], according to the definitions and requirements issued from users, which are also gathered and analysed in the context of the project. The aim is to design a retargetable framework, i.e. a framework easily portable to another execution infrastructure. In order to demonstrate this ability, a porting will be performed during the project.

More precisely, the work related to framework implantation will involve the following:

### ■ Interaction models

In its original shape, the CORBA component model (CCM) offers limited interaction support. In the context of RTE systems, it is necessary to make available an extensive range of highly customisable interaction mechanisms with, for example, quality of service enforcement. This involves adapting the CCM accordingly. A significant amount of work has already been performed on this and some results are available. The core idea resides in adding 'connectors' to the CCM: connectors are software entities dedicated to interaction management [4] that are used to interconnect components. All necessary modifications to the CCM have already been performed to include this new software entity, and a large part of prototyping has been performed. More details on this topic can be found in: 'How ADLs can help in adapting the CORBA component model to real-time embedded software design' [5].

### ■ Container runtime reference implementation

The purpose of this task is to specify and implement the container runtime so that the CCM containers can integrate nicely all the technical properties that are to be provided for the components. Note that some parts of the containers' implementation may depend on the underlying execution platform. The starting point is the lightweight CCM Object Management Group (OMG) specification, and the reference implementation is real-time CORBA based. Guidelines for porting to another infrastructure will also be provided.



## INNOVATION REPORT

### ■ Container Building Support

The CCM approach involves generating containers. First, a description of components is written in a dedicated language – the interface definition language (IDL). Then, this description is used as input to a generation tool which produces the container code. Since MERCED will largely impact the contents of containers to make them specific to real-time support, it is necessary to provide the corresponding generation tools. Note that the generation tool is also part of the retargetable environment: a first version will be provided to generate real-time CORBA-compliant containers, but porting rules will also be issued and a porting performed.

### ■ Predefined components implementation

The framework will be completed by predefined components: these contain processing at the same time specific to a given application domain such as telecommunications and generic – the processing is of interest for all applications of the given domain. These predefined components will be implemented for two application domains: software radio and electrical distribution. As these components are plain by construction, they will benefit – as any application component – from the hosting support provided by the execution platform; their integration will therefore be obvious.

### ■ Configuration and Deployment Suite

The CCM is accompanied by another OMG specification, which provides guidance in application assembly and deployment [6]. ‘Assembly’ is a step in which the application architecture is described in terms of component types and interconnections. This description is based on a dedicated XML format, also defined in the specification. Once assembly is done, a deployment tool is fed with the architecture description and performs the relevant component instantiations and interconnections. All these facilities will be adapted in the scope of MERCED to fulfil RTE-specific requirements and then implemented. Particular attention will be paid to the embedability of the deployment tool.

### ■ Port of the Execution Support on top of OSEK

As said before, the aim is to design an easily portable framework. In order to assess the retargetability of the implemented framework, a porting will be realised to the *Open systems and their interfaces for the electronics in motor vehicles* (OSEK) [7] operating system together with an OSEKcom communication layer [8]. OSEK is a specialised execution infrastructure for highly constrained hardware platforms. One of its main features is to be highly static: the porting from real-time CORBA to OSEK thus promises to be a great technical challenge.

### ■ Use cases

The last step will consist in an assessment of the materials defined and implemented during the project, by means of use cases from various application domains: software-defined radio, electrical distribution and metal processing.

## Conclusions

The MERCED project proposes to compensate the rise in complexity in the RTE domain by enforcing a COTS-based development approach. Such an approach would permit a decrease in development costs, while keeping a sufficient level of reliability. However, in order for COTS to achieve their breakthrough in the embedded systems area, proper tools and methods have to be defined. MERCED proposes an answer to this concern by defining and implementing a component-reuse environment and a retargetable execution framework, both specialised for the real-time embedded domain.



## INNOVATION REPORT

### Authors

Sébastien Gérard, CEA Saclay - DRT/LIST/DTSI/SOL/L-LSP, France (sebastien.gerard@cea.fr)

Ansgar Radermacher, CEA Saclay - DRT/LIST/DTSI/SOL/L-LSP, France (ansgar.radermacher@cea.fr)

Sylvain Robert, CEA Saclay - DRT/LIST/DTSI/SOL/L-LSP, France (sylvain.robert@cea.fr)

François Terrier, CEA Saclay - DRT/LIST/DTSI/SOL/L-LSP, France (francois.terrier@cea.fr)

Anne-Marie Fouillart, Thales Communications, France (anne-marie.fouillart@thalesgroup.com)

Olivier Hachet, Thales Communications, France (olivier.hachet@thalesgroup.com)

Vincent Seignole, Thales Communications, France (vincent.seignole@thalesgroup.com)

Virginie Watine, Thales Communications, France (virginie.watine@thalesgroup.com)

### References

- [1] *CORBA Components version 3.0*, Object Management Group, 2002.
- [2] *Lightweight CORBA Component Model - OMG draft adopted specification*, Object Management Group, 2003.
- [3] *Real-Time CORBA specification version 2.0*, formal/03-11-01, Object Management Group, 2003.
- [4] *Software connectors and their roles in component development*, D. Balek and F. Plasil, DAIS'2001.
- [5] *How ADLs can help in adapting the CORBA Component Model to Real-Time Embedded Software Design*, S. Robert, A. Radermacher, V. Seignole et al., in Proceedings of Workshop on Architecture Description Language, pp. 117-132, Springer, 2004.
- [6] *Specification for deployment and configuration of component-based applications - draft adopted specification*, Object Management Group, 2003.
- [7] *OSEK/VDX Operating System version 2.2.1*, OSEK/VDX, 2003.
- [8] *OSEK/VDX Communication version 3.0.3*, OSEK/VDX, 2004.