ESPRIT Project #25646

Rapid Prototyping Environment for Real-Time Distributed Systems

D1.9 Final Report (Publishable Version)

64004f1

Report Editor: Gail Hall
TRW Automotive
21 February 2000

THIS DOCUMENT IS NON-MAINTAINED UNLESS IDENTIFIED AS A CONTROL COPY DEFINED BY THE CIRCULATION LIST ON PAGE TWO

TRW Automotive

DaimlerChrysler AG

Matra BAe Dynamics France SA

IT-Innovation Centre

Integrated Systems Inc Ltd

Kennedy Carter Ltd

Prover Technology AB

Executive Summary

This report forms the final deliverable for the RAPIER Project (ESPRIT 25646). The project involves the following partners: TRW Automotive (UK), DaimlerChrysler (D), Matra BAe Dynamics France (F), Kennedy Carter (UK), ISI (UK), Prover Technology (S), IT-Innovation Centre (UK).

This report was collated on behalf of the RAPIER project consortium by TRW Automotive. The main points of contact within each of the partner organisations contributing to this report are listed below:

DaimlerChrysler	Joachim Hofmann
IT-Innovation Centre	James Hammond
Kennedy Carter Ltd	Colin Carter
Matra B.Ae Dynamics France	Brigitte Saget
Prover Technology	Marten Saflund
TRW Automotive	Gail Hall

The RAPIER project was executed over a period of 27.5 months, beginning in October 1997. During this time the partners developed 2 demonstrator applications, a development process and an integrated toolset to support this process. The project, whilst it did not achieve all of its objectives, demonstrated a modelling process suitable for real-time embedded control systems which allows the use of 'best of breed' tools and is flexible enough to deal with late changes in processor platform. Automatic code generation from Object Oriented Analysis was achieved and early Timing Analysis, possible with the TASMAN tool developed in the project, was demonstrated as a beneficial process element.

Circulation List	
Executive Summary	Full Report
Dr S Prosser Mr A Kennedy Mr M Foulon	Dr M Rohen (3 copies) Mrs Gail Hall Mr C Carter Dr P Hofmann Ms B Saget Mr M Serughetti Mr M Säflund Mr R McKendrick Original to Website

Table of Contents

1 Introduction 5

2 Technical Report for Publication 6

2.1 Project Objectives 6

2.2 Project Activities and Achievements 7

3 Conclusions and Recommendations 10

Abbreviations

CORBA	Common Object Request Broker Architecture
DC	DaimlerChrysler
Intelligent OOA	OOA tool from KC
ISI	Integrated Systems Inc Ltd
IT-Innovation	IT-Innovation Centre
KC	Kennedy Carter Ltd
MATRIXx	A modelling and development tool from ISI
MBDF	Matra B.Ae Dynamics France
MSFOL	Many Sorted First Order Logic
OOA	Object Oriented Analysis
OSEK	Open Systems and their Corresponding Interfaces for Automotive Electronics ("*Offen System und deren schnittstellen Fur die Electronik im Kraftfahrzeug*")
PERTS	Tools for the analysis, validation and evaluation or real-time systems. From Tri-Pacific software
PT	Prover Technology
SMUG	Shlaer-Mellor[1] User Group
TASMAN	**T**iming **A**nalysis for **S**hlaer-**M**ellor real-time **A**pplicatio**N**s
TRW	TRW Automotive
UML	Unified Modelling Language

[1] Shlaer-Mellor is an OOA methodology

1 Introduction

This report describes the activities and outcomes of the RAPIER Project (ESPRIT 25646). The project involves the following partners: TRW Automotive (UK), Daimler-Chrysler (D), Matra BAe Dynamics France (F), Kennedy Carter (UK), ISI (UK), Prover Technology (S), IT-Innovation Centre (UK). The co-ordinating partner is TRW Automotive.

The RAPIER project began in October 1997 with a planned duration of 24 months, subsequently extended to 27.5 months.

The principal objectives of the RAPIER project were to produce a prototype integrated toolset for the requirements analysis, design, validation and automatic implementation of real-time embedded systems to enable end-users to cope with increasing complexity of applications, increased variability in hardware architectures and pressure to reduce development times.

The partners to the project represent both the automotive and aerospace industries, including suppliers and end-users. Tool vendors and the academic world are also well-represented. This spread of partners was selected to give maximum potential for a successful outcome that could be exploited at the end of the project. The work identified within the project was split into ten workpackages to which the partners contributed to according to their expertise.

2 Technical Report for Publication

2.1 Project Objectives

System development in the automotive and aerospace industries must become more productive. Complexity increases rapidly with each generation of system and market pressures demand shorter development times. Current practices are already stretched and will not cope in the future. These industries report that the range of techniques and tools adopted is very narrow with the result that techniques and tools are often used for inappropriate tasks and that some beneficial activities are not performed.

Isolated experience within the industries has shown considerable value can be gained from:

- Executable object modelling
- Mathematical and control modelling
- Timing analysis
- Safety verification
- Target code generation

None of these techniques are new. They are all still developing but have sufficient maturity to be candidates for integration. The objective of the RAPIER project is to bring these techniques and their supporting tools together to provide a coherent process supported by an integrated 'best-of-breed' toolset. This will enable the industries to:

- Select the best tools for the job at hand
- Cope with increasing complexity
- Shorten development lifecycles by early verification
- Shorten development lifecycles by the reuse of intellectual property
- Shorten development lifecycles through the extensive use of optimised code generation
- Use valuable techniques not previously used
- Manage manufacturer and supplier parallel development activities
- Reduce the maintenance overhead by eliminating redundant information in the development process

2.2 Project Activities and Achievements

Executable object modelling based upon Shlaer Mellor Object Oriented Analysis provides system partitioning, abstraction and precise modelling techniques. Executable models can be run in a simulation environment to verify their behaviour. Mathematical and control modelling techniques are widely used in the automotive and aerospace industries and share the capability of providing observable model execution in a simulation environment. Executable object modelling does not have specialised mathematical modelling capabilities built-in and conversely mathematical modelling does not strongly support system partitioning and abstraction. The two techniques are therefore complementary and the execution semantics are such that combined simulations are possible.

The project succeeded in integrating executable object modelling with mathematical modelling using Intelligent OOA and MATRIXx as the tools of choice. Co-simulation using MATRIXx models as embedded algorithms in the object models, and as plant models, was proven with the demonstrators.

The execution semantics of the models mean that information can be extracted from the models which can be used for other purposes including timing analysis and safety verification.

The project developed TASMAN, a tool capable of extracting timing information from the executable object models and making it available in a suitable format for the PERTS[2] tool, a commercially available schedulability analysis tool.

Although timing analysis had been used in the industries, it was generally either a separate parallel task, which would lead to inconsistencies between the development models and timing analysis information, or performed late in the development cycle, which meant problems found were likely to be difficult and expensive to fix.

TASMAN provides a tool and process for performing timing analysis. TASMAN is closely integrated with Intelligent OOA, providing the developer with a smooth development environment. Since executable models can be available early in the development cycle, it is possible to start timing analysis early too.

The project investigated the feasibility of extracting sufficient information from the object models to make safety analysis possible. It was concluded that a subset of the object modelling formalism could be identified which would allow formal verification. The identified subset would have been sufficiently rich to model safety related parts of the system. However, there was insufficient time or resource to implement the translation from OOA object models to Many Sorted First Order Logic (MSFOL). The project therefore failed in its objective to integrate formal verification as part of the toolset.

Executable models are also suitable as a basis for code generation. However, the demands of the industries require that hardware architectures are typically restricted. This means that a simplistic approach to code generation is unsuitable. The RAPIER project developed an optimised code generator aimed at a small range of target architectures, as

[2] PERTS is a product from Tri-Pacific software (http://www.tripac.com)

represented by the demonstrators. The RAPIER code generator was based upon a tailoring of the commercially available Intelligent Configurable Code Generator.

The RAPIER code generator was used to translate executable object models into C code running in the target architectures of the demonstrators. The automotive demonstrator using an OSEK Real Time Operating System and the aerospace demonstrator using a Matra supplied proprietary Real Time Operating System. The mathematical and control models are translated into target code using the MATRIXx Autocode capability. In addition the automotive demonstrator was applied to a Siemens 167 processor to prove the portability of an application to an alternative hardware platform at a late stage in development.

In summary, system development using the RAPIER process covers:

- requirements capture
- system partitioning into domains
- selection of an object modelling or control modelling approach for each domain
- object modelling selected domains
- mathematical and control modelling of other domains
- specification of the tasking model
- simulation and testing of single domains
- integration, co-simulation and testing of multiple domains
- performing model-based timing analysis (estimates)
- generation of code for target architectures
- performing target-based timing analysis (measurements)
- target based testing

A list of activities creates the impression of a linear process, though the process is executed in an iterative and incremental fashion and that the dependencies are such there is plenty of scope for parallel working.

The demonstrator projects followed the emerging process and used the emerging toolsets to demonstrate the practical application of the process and tools on representative systems. The process was captured in a process definition compiled by the project.

The DaimlerChrysler Research and Technology Department chose to evaluate and assess the RAPIER process and toolset using an application from their commercial vehicle department. The so-called Ride Height application is a digital two point control system with the major mission being to keep the vehicle at a constant stable level in situations like loading, unloading and driving. DaimlerChrysler, in cooperation with TRW, used the provided seamless toolset supporting distributed development to develop the autocoded

application software running on an OSEK compliant OS and DaimlerChrysler specific prototypical hardware platform (PowerPC).

The aerospace demonstrator developed in RAPIER for the purpose of evaluating the methodology and tool-set, was based on a simplified auto-pilot function. The evaluation work performed by MBDF included evaluation of the co-simulation benefits from using tools targeted to designing specialised domains. Specifically MBDF used:

- Modelling of the generic structure of a Guidance & Control application, using I-OOA
- Modelling of the control algorithms, using MATRIXx (these models are encapsulated by the services defined in the I-OOA models)
- Modelling of the system environment, in MATRIXx.

The effectiveness of automatic generation of code from both models (MATRIXx, I-OOA) and the use of a timing analyser (TASMAN) working from the models were also evaluated as part of the aerospace demonstrator.

The importance of standards compliance was recognised by the project and steps were taken to move the RAPIER toolset towards standards compliance wherever possible. The sstandard Unified Modelling Language (UML) notation is fully satisfied by the RAPIER toolset and the project is directly involved with the on going executable UML standardisation process. Current RAPIER executable object models will be compatible with the Executable UML standard that emerges in late 2000. As part of exploitation, it is intended that the commercial version of the RAPIER toolset will use the CORBA standard as the basis for the co-simulation backplane allowing multiple tools to be integrated in a co-simulation environment.

3 Conclusions and Recommendations

Co-simulation using specialised design tools allowed the designers to use the best tool for each domain, providing validation capabilities for the overall application in a closed-loop environment. Co-simulation permitted functional validation to take place prior to any implementation, with TASMAN used to indicate any timing issues. This co-simulation provides a reference for subsequent validation as the development proceeds.

The Aerospace demonstrator found that automatically generated code provided excellent traceability between models and code. Interestingly it was found that code optimisation is related to the modelling choices made, thus it is important for the designer to consider the performance required in the early stages of modelling. Whilst the code generation was not sufficiently developed during the project to permit its use for production code for high volume manufactured applications, it is suitable for functional validation, feasibility studies requiring rapid results, demonstrators and other applications which do not require fully optimised code. Further optimisations have been identified which it was not possible to implement during the project itself but which nevertheless continue to be developed outside this programme.

Timing Analysis on the aerospace demonstrator showed benefits in obtaining early overall estimates of the dynamic behaviour of the application. This provides an important input to the considered selection of an architecture since trade-offs between cost and performance become more informed.

The automotive demonstrator provided an opportunity to use domain partitioning as a basis for distributed development. Domains were allocated to DaimlerChrysler and to TRW Automotive for development prior to integration. This showed that it is possible to use development teams located in different locations, but that great care is needed in the definition of these domains at the start of a project.

The project developed a coherent process definition and took account of ongoing standards development work. Alongside the tool development activities, this forms the basis for ongoing work within the various partners. There are active plans to take parts of the toolset forward to the marketplace and whilst the end-user partners may not adopt the process in its entirety in the short-term, there is sufficient interest in parts of the process and toolset to consider that the project will influence future ways of working.