IST-004527 ARTIST2
Network of Excellence
on Embedded Systems Design

Activity Progress Report for Year 4

JPRA-Cluster Integration

Real-Time Languages

Clusters:

Adaptive Real-Time

Activity Leader:

Professor Alan Burns, University of York, UK

http://www-users.cs.york.ac.uk/~burns/

Policy Objective (abstract)

To develop flexible real-time systems requires a number of tools and techniques. One of the most important being the programming language used to develop the application code. This activity considers real-time programming languages, both industrial strength and research-based languages are considered. This year has mainly focused on the production of a survey of available languages; this survey is now available on the ARTIST web site. Other work has focussed on the new Ada 2005 standard. An assessment of the expressive power and ease of use of the many new features within that language is currently been undertaken. Other languages within scope of this activity are the RTSJ (Real-Time Specification for Java) and C (with POSIX), along with research-oriented languages.

Table of Contents

1. Overview of the Activity

This activity started in the second year and is concerned with the delivery of flexible systems by the use of appropriate programming language abstractions. The emphasis of the activity has been to bring together language designers from within and beyond ARTIST2 to influence the development of both engineering languages such as Ada and the RTSJ (Real-Time Specification for Java) and research languages.

The main focus this year has been on the production of a wide survey of over 20 languages used for programming real-time systems. This survey is completed and is available on the ARTIST web site. Also this year, work has continued on Ada 2005 (both implementation and pattern development) and the RTSJ (pattern development). Meetings on Ada have been held. A major workshop is planned for next year.

1.1 ARTIST Participants and Roles

Professor Alan Burns – University of York (UK)

The York team is one of the language designers for Ada and the RTSJ (Real-Time Specification for Java). Scheduling expertise for program structures such as budget controllers and servers. Research into general language design for real-time systems. Other research topics covered at York include scheduling, wireless sensor-nets, FPGA implementation and WCET (Worst-Case Execution Time) analysis.

Professor Michael Gonzalez-Harbour – University of Cantabria (Spain)

Contributions to, and evaluation of, the proposed extensions to Ada and the use of Ada 2005. Is undertaking prototype implementation of the proposed extensions to Ada using their MaRTE operating system. Coordinator of the work on POSIX (i.e. extensions for real-time programming).

Professor Juan Zamorano – UP Madrid (Spain)

Contributions to, and evaluation of, the proposed extensions to Ada, Java and POSIX. The UPM team is undertaking a prototype implementation of the new real-time features in Ada 2005 using an evolved version of its Open Ravenscar real-time Kernel (ORK).

Professor Miguel Pinho – Polytechnic Institute of Porto (Portugal)

Contributions to, and evaluation of, the proposed extensions to Ada and Java. Support for kernel monitoring and control. Other research topics include support for server-based scheduling, and dynamic quality of service.

Sergio Yovine – VERIMAG

This group worked on region-based scoped-memory management in RTSJ, mainly in automatically computing regions and their size by program analysis. It also evaluated industrial RTSJ implementations and case studies.

1.2 Affiliated Participants and Roles

Marisol García-Valls, Universidad Carlos III de Madrid (Spain)

The Distributed Real-Time Systems Lab led by Marisol García-Valls works *on the RTSJ (Real-Time Specification for Java), evaluating it, identifying the drawbacks it presents, and proposing extensions and solutions to overcome them. Mainly worked*

on the memory model of the RTSJ and on the introduction of predictability in Java RMI. Other research topics covered at UC3M are real-time middleware, QoS resource management architectures and algorithms, and real-time systems modelling.

1.3 Starting Date, and Expected Ending Date

This activity officially started on March 1st 2006 and continued to the end of the NoE. The issue of language design and use will extend beyond the life of the NoE. New platform configurations (e.g. multi-core SoCs) and application needs (e.g. from the domain of cyber-physical systems) will continue to set new requirements for programming languages.

1.4 Baseline

The Ada language is still in use in many application domains, in particular the safety-critical areas such as avionics and railway signalling. The definition of the language itself has gone through a number of versions; the latest being Ada 2005. In the first standard the support for real-time embedded systems was weak with the concurrency model having a number of limitations. The Ada 95 version was a considerable improvement and did include a well defined set of primitives for undertaking fixed priority (i.e. essentially static) scheduling for non-adaptive applications. The research community, including members of the ARTIST2 community, has been involved in defining new language features that could extend the applicability of Ada, especially to the adaptive (more dynamic) domain of applications. Many of these features have found themselves incorporated into Ada 2005 (again due to the efforts of ARTIST2 members).

Ada 2005 is now defined and has undergone international standardisation. Currently implementation of all the real-time features of the language is awaited. It is appropriate therefore to assess the languages' expressive power in terms of the ease with which it will support the programming of flexible real-time systems. Much of the expertise surrounding Ada now lies within Europe, it is therefore important to build upon this situation to ensure the continuation of this lead. This will involve work within Europe and participation in international events, particularly in the US.

Supporting real-time functionality via language constructs rather than OS calls eases the programmer's task when writing complex applications. ARTIST2 partners have been involved in a number of standardisation activities and in ongoing research into real-time language primitives and associated analysis techniques. ARTIST2 provided the framework for this broad set of activities to compare outcomes and to influence each other's research. One particular area in which ARTIST2 partners are involved is the effort surrounding the use of Java as the core language on which real-time abstractions are built.

1.5 Problems Tackled in Year 4

The main focus of this activity has been the production of a survey of programming languages used in real-time system engineering. The survey has been very wide ranging and covers both research and engineering languages, and languages of different types (i.e. imperative, functional, synchronous, model-based, and platform-based).

The other focus has been on continuing the work undertaken on Ada. As indicted last year, the reason for this focus is that Ada 2005 uniquely has a number of features and abstractions that allow flexible real-time systems to be programmed. These features include EDF scheduling,

CPU time monitoring and control, timing events and the primitives from which execution-time servers can be constructed. Ada 2005 also has incorporated interfaces that make it easier to program reusable utilities (for example time-bounded event handlers). The focus of the Ada 2005 work has thus been the development of a collection of real-time utilities. The research question being addressed is whether the potential for expressive power and reuse that Ada provides can be realised in practice.

Another Ada 2005 problem tackled during this year has been to continue the implementation of run-time support features for the new real-time features of the language. The final service to be implemented has been “Group Budgets”. The implementation has been carried out as a joint effort between AdaCore, a leading Ada compiler developer company, and the University of Cantabria. For this new service different implementation alternatives were studied and their performance was evaluated. As a conclusion of the results obtained the new service can be implemented in an efficient way in an existing run-time system without requiring major changes.

Real-Time Java is an evolving technology. Over the last few years implementations have begun to appear and have now matured. However, there are several problem areas that still need to be addressed. The following were identified during the 2nd year of ARTIST2 and have formed a focus for the work during the third and final year:

1. Community Building — the bringing together of various international research activities
2. Maintaining Momentum — ensuring Real-Time Java continues to evolve to ensure its industrial relevance. This work can be broken down into the following topics:
   - international standards work within the Java Community Process
   - research activities undertaken by the ARTIST partners

1.6 Comments From Year 3 Review

1.6.1 Reviewers’ Comments

“ACCEPTED

This is a quality document. No specific remarks.

2. Summary of Activity Progress

2.1 Previous Work in Year 1

This activity did not start in Year 1.

2.2 Previous Work in Year 2

The activity started towards the end of Year 2. Effort was focused on an initial study of Ada 2005 and its implementation. Also work has been done to plan a series of activities concerned with the development of higher level abstractions for Ada. Part of this work involves work with international collaborators. A meeting/workshop on Ada 2005 was held in York as was an international open workshop on SCOOP.

Ada 2005 has a number of facilities that could make the programming of adaptive real-time systems much more straightforward and therefore likely to be used in an industrial context. But many of these features are new and have not been tested – in the sense of being used in an integrated way to build high-level abstractions. Work has started on this verification, and will continue in the following year.

2.3 Previous Work in Year 3

The main focus in year 3 was again the Ada language, run-time support for the new language features (in Ada 2005) were planned (and partially implemented) and a library of reusable templates was instigated. A number of meetings were held to design the basic structures for these templates and to identify the main abstractions that needed to be supported. Work was also undertaken on implementing Ravenscar, a subset of Ada that is aimed at the safety critical domain.

The other language featured in the third year was Java and the Real-Time Specification (extensions) for Java (RTSJ). Work was undertaken on the memory management features and on extensions aimed at distributed applications. Again a number of key meetings and workshops were organised.

2.4 Current Results

2.4.1 Technical Achievements

Production of a Survey of Programming Languages (York, plus a large number of other ARTIST partners and non-ARTIST individuals linked to the ARTIST community).

The production of real-time and embedded systems involves the use of many different tools and techniques. As these systems become more software centric, programming languages employed in the production of this software are now of crucial importance. Any language has the dual role of enabling expression whilst at the same time limiting the framework of concept and abstractions within which that expressive power may be applied. If a language does not

support a particular notion then programmers cannot apply it and may even be totally unaware of its existence. For real-time programming languages there are many such concepts that are supported to a greater or lesser extent in a range of languages that purports to be appropriate for the embedded systems domain. For example: time, clocks, concurrency, deadlines, events and signals, exceptions, periodicity, scheduling and predictability are all important notions that programmers may wish to address and which should therefore be available to them via the implementation languages that they can employ.

This survey considers over twenty programming languages. The short reports available on each language aim to introduce the main features of the language, provide the links to further sources of information, and give an indication of the current developments within the language. Some languages are mature, used widely and are the subject of rigorous standardisation procedures. Others are research languages, with a small user population and an informal definition. All languages covered in the survey are implementation languages in that they are supported by tools (typically compilers) which generate executable code for the designated hardware platform. To give a structure to the survey each language is placed in one of five classes: imperative, functional, synchronous, model-based, and platform-based. However this is a loose classification as some languages could easily be placed in more than one category, and imperative languages can usually be constrained to mimic the other styles.

The survey in on the public ARTIST web site (follow links from home page, dissemination and Contribution to Standards) – or use the following.

http://www.artist-embedded.org/artist/ARTIST-Survey-of-Programming.html

The following have made contributions to this survey. Alejandro Alonso, Pierre Boulet, Frédéric Boussinot, Christian Buckl, Alan Burns, Magnus Carlsson, Paul Caspi, Jorge Coelho, Henk Corporaal, Susan Davidson, Víctor Fernández, Sebastian Fischmeister, Oana Florescu, Andy Gill, Thierry Gautier, Marc Geilen, Sébastien Gérard, Paul Le Guernic, Nicolas Halbwachs, Michael González Harbour, Leandro Soares Indrusiak, Julia L. Lawall, Insup Lee, Per Lindgren, Pieter Mosterman, Gilles Muller, Johan Nordlander, Luís Miguel Pinho, Juan Antonio de la Puente, Bran Selic, Robert de Simone, Bjorn von Sydow, Jean-Pierre Talpin, François Terrier, Bart Theelen, Stavros Tripakis, Marisol Garcia Valls, Eugenio Villar, Jeroen Voeten, Andy Wellings, Reinhard Wilhelm, Victor Wolf and Sergio Yovine.

Development of Real-Time Utilities for Ada 2005 (Cantabria, Porto, UP Madrid, York)

The work started in year 3 is continuing with the development of patterns for real-time utilities. One aspect that was recognised during the second year was the limitation of the Ada 2005 model in that it did not allow requeue via interfaces. A report of this drawback was forwarded to the Ada language team and as a result the definition of Ada 2005 has been changed. This change has now been implemented by the main compiler team and is available.

Progress has not been as extensive as indicated last year by the initial lack of support for one of the key features of Ada 2005, namely group budgets. The current situation is that the feature is now available (see below), but extensive experience in using the feature has not yet occurred. This will occur in the future.

Finally, to note, extensions of the utilities to include support for mode changes has been progressing and will be reviewed at the next IRTAW in 2009.

A new technology has been developed for developing distributed and real-time component-based applications in Ada. The technology uses the “interfaces” mechanism, recently added in Ada 2005 to support multiple inheritance. Tools have been implemented to support the technology by automatic code generation of the elements that are needed to provide communication among parts of the application that are deployed in different processing nodes. This communication takes into account the real-time requirements specified in the deployment and configuration plan of the application. The approach allows full control of real-time properties both in the processors and in the networks, while eliminating the need of using a middleware layer.

Two middleware implementations providing support for distributed applications in Ada have been extended to support real-time requirements, both in the processing nodes and in the networks. The first of these implementations, called PolyOrb, supports different distribution technologies; among them, CORBA, and the Distributed Systems Annex (DSA) of the Ada specification. The second implementation is called Glade and uses the DSA technology. A comparison among the different implementations has been made providing information that is useful to decide under which circumstances one implementation offers advantages over the other one.

Implementation of Ada 2005 (Cantabria)

AdaCore, a leading Ada compiler vendor that develops and distributes the gnat free software compiler, now distributes the MaRTE operating system developed at the University of Cantabria, as one of the run-time systems that provides underlying support to the concurrency of Ada tasks. This run-time system, available on Linux platforms, offers all of the features defined in the real-time Annex of Ada 95, and some of the features defined in Ada 2005. In the current reporting period work has continued to include in this implementation the remaining Ada 2005 services. Implementation of task group budgets has now been completed. In addition, implementation efforts have been made to better integrate MaRTE OS with the gnat run-time system.

Ada 2005 real-time mechanisms (UP Madrid, York)

The UPM team have been working on implementing the Ada 2005 real-time mechanisms on the Open Ravenscar real-time Kernel (ORK), aimed at high-integrity systems using the Ravenscar profile of Ada tasking on SPARC 8 (LEON) computers. A new version of the kernel, ORK+, has been developed, which includes execution-time clocks and timers, timing events and group budgets. Since execution-time timers and group budgets are not allowed in the Ada 2005 definition of the Ravenscar profile, an extended profile has been defined that includes up to one execution-time timer per task, declared at library level. ORK+ is a core component of the ASSERT Virtual Machine, the execution platform for real-time components developed in the ASSERT project.

The kernel is integrated with the GNAT GPL 2007 compilation system, and can be downloaded from

http://www.dit.upm.es/ork

Real-Time Java Community Building (UC3M, York)

The real-time Java user (academic and industrial) community is still fragmented. However, the series of Real-Time Java Workshop (JTRES – Java Technology for Real-time and Embedded

Systems), is now firmly established. The 2007 workshop was held in Vienna and the 2008 one at Sun Microsystems Santa Clara campus. York has played a leading role on the steering committee for these workshops.

Within the ARTIST partners, the main Java-related research work is being done by UC3M and York. UC3M is working on the distributed version of RTSJ; from the implementation of a real-time RMI, the group has identified, specified, and implemented over the past years a number of extensions to RTSJ. These extensions are a means of facilitating the implementation of real-time distributed applications based on RTSJ by transferring to the middleware (DRTSJ) the job of guaranteeing real-time interaction. York is addressing a range of issues including asynchronous event handling, mobile code, multiprocessor issues.

Maintaining Momentum: International Standards Work (York)

The main implementation approach of real-time Java, the Real-Time Specification for Java (RTSJ) continues to evolve. Over the last year, the main work has been undertaken in two Java Specification Requests (JSR 282 and JSR 302).

JSR 282 is, among other things, trying to fill some of the gaps that were left in the original specification, and to begin to explore more direct supports for SMPs. Proposed enhancements range from changes in scheduling models to improved memory management issues to offer better support to its region model. This process is going to be useful not only for programmer but for other specifications which will act as the acid test for RTSJ. JSR 302 is considering safety and mission critical profiles. York plays a major role in both these activities.

Distributed real-time Java technologies require an extraordinary effort in order to integrate the current centralized real-time Java languages with the traditional middleware distribution paradigms. JSR 50 leads this initiative; however, currently, there the work on the distributed version of RTSJ is stopped and no newer drafts have been released since 2006. Considering that distribution will become a key in the success of Java-based distributed applications with real-time requirements, the work carried out at UC3M is aimed at the construction of an architecture which offers platform independence (see below).

Maintaining Momentum: Research Work (UC3M, Verimag, York)

This UC3M group has focused on Distributed Real-Time Java Issues. The work provides a set of important abstractions such as support for a distributed garbage collection and a naming service, more related to the RMI (Remote Method Invocation) model. Also, it contains other contributions like the possibility of using asynchronous remote invocations or centralized synchronization services which are useful in the development of many real-time systems. An initial programming interface, called DREQUIEMI, for this computation model has been already defined; it is aligned with the RMI (Remote Method Invocation) and the RTSJ (Real-time Specification for Java) technologies. Besides, in the specific context of the RTSJ, a set of extensions (to the region model, to the reference model and to the threading model) has been proposed to simplify the development of both, centralized and distributed real-time applications. In the course of the last year, the group of Carlos III University of Madrid has carried out suite calculations and measurements to determine the drawbacks in communication when using the complex layer structure of Sun's Java RMI. This practical side identifies room for improvements in the remote invocation theoretical model, current implementation, and mismatches on the integration of RMI and RTSJ to become DRTSJ.

From the perspective of memory management, the results of UC3M on the implementation of parts of DRTSJ, especially on memory management issues, link the work of the group with JCP-282. UC3M has worked on the evaluation of current programming extensions designed to violate assignment and single-parent rules that are insufficient for the demands of a distribution middleware. Current violation mechanism, named portals, cannot be accepted as valid mechanism to gain access to an arbitrary region; an extension to the current programming interface is required in order to offer more general facilities which make such a process easier. Furthermore, this enhancement has to extend not only the weak semantics of Java (now considered as an item in the JCP-282 list), but also the case of strong references, which now are forgotten by the JCP-282. The second is the need of other region models which offer improvements to the current region model, thus giving some kind of extra support for specific demands. For instance, current infrastructures do not allow to recover regions automatically once they are empty, one mechanism which may be useful to generate a valuable support for no-heap distributed remote objects in DRTSJ. UC3M has also further elaborated the concurrency model of RTSJ. The group's experience in DRTSJ implementation demonstrated the usefulness of having a more flexible threading infrastructure for RTSJ.

Java-based distributed real-time systems require appropriate memory management. Since currently no 100% real-time garbage collection is possible, predictable memory management relies mostly on its avoidance in favor of using memory regions. Currently, RTSJ offers a region-based model for the development of centralized systems, but its distributed equivalents lack a similar support. The group of Carlos III University of Madrid has developed the "no-heap remote object" paradigm that allows to remove the objects created during a remote invocation automatically. The model extraordinarily reduces the number of changes required in the platform to support it. On the other hand, VERIMAG worked on the direction of trying to provide predictable dynamic memory management. For this, the group developed a static analysis capable of providing a safe but tight approximation of the peak memory occupancy required to run a program using RTSJ scoped-memory allocation.

The EU JEOPARD project has recently been started with a remit to investigate SMP and multicore issues. York is a member of the team. In order to make the RTSJ fully defined for SMP multiprocessor systems, the following issues have been addressed.

- The dispatching model -- the current specification has a conceptual model which assumes a single run queue per priority level.
- The allocation model -- the current specification provides no mechanisms to support processor affinity,
- The synchronization model -- the current specification does not distinguish between synchrononized methods which suspend holding their locks and those that do not (this is important for multiprocessor synchronization algorithms),
- The cost enforcement model -- the current specification does not consider the fact that processing groups can contain scheduling objects which might be simultaneously executing,
- The affinity of interrupts (happenings) -- the current specification provides no mechanism to tie interrupts (happenings) and their handlers to particular processors, and
- The failure model -- the current specification makes no statements about partial failures of the underlying platform.

2.4.2 Individual Publications Resulting from these Achievements

York

Mohammed ALRahmawy, Andy Wellings (2007), A model for real time mobility based on the RTSJ, JTRES '07: Proceedings of the 5th international workshop on Java technologies for real-time and embedded systems.

A. Burns, A.J. Wellings (2007), Delivering Real-Time Behaviour, Domain Modeling and Duration Calculus, Springer, LNCS 4710.

MinSeong Kim, Andy Wellings (2007), Asynchronous event handling in the real-time specification for Java, JTRES '07: Proceedings of the 5th international workshop on Java technologies for real-time and embedded systems.

MinSeong Kim, Andy Wellings (2008), An efficient and predictable implementation of asynchronous event handling in the RTSJ, JTRES '08: Proceedings of the 6th international workshop on Java technologies for real-time and embedded systems.

Osmar M. dos Santos, and Andy Wellings (2007), Cost enforcement in the real-time specification for Java. Real-Time Systems, 37(2).

Osmar M. dos Santos, and Andy Wellings (2008), Run Time Detection of Blocking Time Violations in Real-Time Systems, 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications.

Osmar Marchi dos Santos and Andy Wellings (2008), Blocking time monitoring in the Real-Time Specification for Java, JTRES '08: Proceedings of the 6th international workshop on Java technologies for real-time and embedded systems.

A.J. Wellings (2008), Multiprocessors and the Real-Time Specification for Java, Proceedings of the 11th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing ISORC-2008.

A.J. Wellings (2008), Processing Group Parameters and the Real-Time Specification for Java, JTRES '08: Proceedings of the 6th international workshop on Java technologies for real-time and embedded systems.

Universidad de Cantabria

Patricia López Martínez, José M. Drake, Pablo Pacheco, and Julio L. Medina. "An Ada 2005 Technology for Distributed and Real-Time Component-based Applications." 13th International Conference on Reliable Software Technologies, Ada-Europe, Venice (Italy), in Lecture Notes on Computer Science, Springer, LNCS 5026, June, 2008, ISBN: 3-540-68621-7, pp. 254-267.

Héctor Pérez, J. Javier Gutiérrez, Daniel Sangorrín, and Michael González Harbour. "Real-Time Distribution Middleware from the Ada Perspective". 13th International Conference on Reliable Software Technologies, Ada-Europe, Venice (Italy), in Lecture Notes on Computer Science, Springer, LNCS 5026, June, 2008, ISBN: 3-540-68621-7.

Universidad Politécnica de Madrid

Santiago Urueña, Juan Zamorano, José A. Pulido, Juan A. de la Puente.
Communication Paradigms for High-Integrity Distributed Systems with Hard Real-Time

Requirements. In Bernd Kleinjohann, Lisa Kleinjohann, Wayne Wolf (eds.), *Distributed Embedded Systems: Design, Middleware and Resources*. Springer, Boston, 2008.

Juan A. de la Puente, Juan Zamorano, José A. Pulido, Santiago Urueña. The ASSERT Virtual Machine: A Predictable Platform for Real-Time Systems. In Myung Jin Chung, Pradeep Misra (eds.), Proceedings of the 17th IFAC World Congress. IFAC-PapersOnLine, 2008.

Santiago Urueña, José A. Pulido, Jorge López, Juan Zamorano, Juan A. de la Puente. A New Approach to Memory Partitioning in On-board Spacecraft Software. In Fabrice Kordon and Tullio Vardanega (eds.), *Reliable Software Technologies — Ada-Europe 2008*, LNCS 5026, pp. 1–14. Springer-Verlag, 2008.

Juan Zamorano, Juan A. de la Puente, José A. Pulido, Santiago Urueña. The ASSERT Virtual Machine Kernel: Support for preservation of temporal properties. *Data Systems in Aerospace — DASIA 2008*.Palma de Mallorca, Spain, 2008.

Universidad Carlos III de Madrid.

P. Basanta-Val, García-Valls, M., and Estévez-Ayres, I. 2008. Simplifying the Dualized Threading Model of RTSJ. In Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (Isorc) - Volume 00 (May 05 - 07, 2008). ISORC. IEEE Computer Society, Washington, DC, 265-272.

P. Basanta-Val, M. García-Valls, I. Estévez-Ayres y J. Fernández-González .*Integración de capacidades de multiplexación en el conjunto de subprotocolos JRMP*. To be Published in *IEEE* América *Latina*, *ISSN* 1548-0992. In Spanish. 2008.

Iria Estévez Ayres, Marisol García Valls, Luís Almeida, and Pablo Basanta-Val. *Solutions for Supporting Composition of Service-Based Real-Time Applications* Proceedings. of the 11th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing, ISORC 2008. Orlando, Florida, USA. May 5 - May 7, 2008.

VERIMAG

V. Braberman, F. Fernandez, D. Garbervetsky, S. Yovine. Parametric Prediction of Heap Memory Requirements. In Proc. of International Symposium on Memory Management (ISMM'08), June 7-8, 2008, Tucson, Arizona, USA. ACM 2008.

2.4.3 Interaction and Building Excellence between Partners

The Ada work has progressed over a number of years and involves close interactions between Cantabria, Porto, UP Madrid and York (and other non ARTIST2 European colleagues). Meetings such as the IRTAW series are collectively organised by members of these groups and these meetings and other ARTIST2 meetings enable members to meet regularly. Visits to each other's sites take place and interactions via email allow the work on patterns to be progressed. Key to this interaction is the development of implementations of the new features of Ada 2005. These are being undertaken by Cantabria and UP Madrid with Porto and York acting as beta test sites.

The work on Java involves three sites: UC3M, VERIMAG and York. Again electronic discussion takes place and meetings such as at JTRES. Wellings from York is a member of the expert group with responsibility for the RTSJ; he is able to bring forward ideas and proposals

from the other partners to the standardisation process. Burns (also from York) fulfils a similar function for Ada (he is a member of the ARG and WG9 – the ISO committees that are responsible for the Ada standard).

Partners are also members of other EU projects (for example FRESCOR). These projects facilitate close collaboration and have regular meetings in which broader issues such as programming languages are discussed.

2.4.4 Joint Publications Resulting from these Achievements

A number of publications acknowledge the help and support of other members of ARTIST2. Workshop papers for IRTAW and JTRES for example usually have single site authorship, but they set the context for joint discussions and future work. There are however some joint publications:

M. Masmano, I. Ripoll, J. Real, A. Crespo, A. J. Wellings (2008), Implementation of a constant-time dynamic storage allocation, Software: Practice and Experience, 38(10).

M. Prochazka, R. Ward, A. J. Wellings (2007), A First Step towards using real-Time Java for Spacecraft On-Board, Data Systems in Aerospace (DASIA 2007), ESA SP-638.

2.4.5 Keynotes, Workshops, Tutorials

Michael González Harbour and Mario Aldea Rivas. "The MaRTE OS run-time as a support platform for real-time programming in Ada 2005". Invited talk. Ada UK Conference, Manchester, September, 2007.

3. Milestones and Future Evolution Beyond the NoE

3.1 Future Milestones

- Publish via a web site a set of real-time programming patterns (repository) for use by Ada 2005 application programmers.
- Complete the implementation of all the Ada 2005 real-time features, in particular the support for EDF scheduling.

3.2 Indicators for Integration

- Joint involvement in planning and participating in a number of workshops.
- Visits between sites.
- Joint commitment to construct and populate a web-based repository of Ada 2005 patterns for real-time algorithms and structures.
- The development of implementations for Ada 2005 that will be tested at other sites.
- Continued work on the specification and evaluation of the RTSJ.
- Further work on the white paper on programming languages for real-time systems.
- Involvement in the planning of workshops, in particular the next IRTAW.

3.3 Main Funding

In addition to the specific funds from the ARTIST2 NoE, the main sources of funding are:

- JEOPARD – EU funded project on parallel real-time Java, consortium includes Vienna and York.
- HIJA – High Integrity Java – EU funded project in which the following ARTIST2 partners are involved: University of York, University of Madrid.
- Javamen – Java on FPGA platforms – UK national DTI project involving the University of York
- Reflect - Reflection Mechanisms in Real-Time Embedded Systems – Portuguese funded FCT project involving the Polytechnic Institute of Porto.
- CooperatES - QoS-Aware Cooperative Embedded Systems – Portuguese funded FCT project involving the Polytechnic Institute of Porto.
- RESCUE - REliable and Safe Code execUtion for Embedded systems – Portuguese funded FCT project involving the Polytechnic Institute of Porto.
- THREAD Spanish project, in which the following ARTIST2 partners are involved: Technical University of Madrid, University of Cantabria, Technical University of Valencia.
- FRESCOR – EU funded project on Flexible scheduling.

- ASSERT - Automated proof based System and Software Engineering for Real-Time applications. EU funded project. Main objective is to improve the system-and-software development process for critical embedded real-time systems in the Aerospace and Transportation domains.
- AdaCore: A contract has been established between AdaCore and the University of Cantabria to complete the implementation of Ada 2005 real-time services using MaRTE OS as a support platform for the GNAT run-time system.
- DYNAMO: French funded in which VERIMAG was involved for the analysis of real-time properties of RTSJ programs dynamic thread creation and memory allocation.
- MADEJA: French funded (Region Rhône-Alpes) in which VERIMAG was involved for studying quantitative prediction of dynamic memory allocation for RTSJ scoped-memory mechanism.

4. Internal Reviewers for this Deliverable

Michael González Harbour (Universidad de Cantabria)

Peter Puschner (Technische Universitaet Wien)

Andy Wellings (University of York)