IST-004527 ARTIST2
Network of Excellence
on Embedded Systems Design

Activity Progress Report for Year 3

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. Current focus has been 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), C (with POSIX), SCOOP (a concurrent real-time extension to Eiffel) and Hume (a functional language).

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 such as SCOOP (a collection of possible extensions to Eiffel) and Hume (a functional language).

The main focus this year has been on the new Ada 2005 standard and the RTSJ. Two major workshops have been held. Planning of a number of activities for the coming year continues.

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 topic covered at York include 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 Opern 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 Yovin – 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. It is envisaged to run to the end of the NoE. The issue of language design and use is likely to 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 easy 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 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 efforts surrounding the use of Java as the core language on which real-time abstractions are built.

Problem Tackled in Year 3

The main focus of this activity is the new Ada 2005 language definition. 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. The focus of the Ada 2005 work has thus been the development of a collection of real-time utilities (see discussion on achieves in section 2.3.1). 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 the implementation of run-time support features for the new real-time features of the language. The implemented services have been “Execution Time Clocks”, “Execution Time Timers”, “Timing Events” and “Dynamic Priorities for Protected Objects”. Low-level support has also been developed for “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. The first part of this effort has been the adaptation of the MaRTE OS kernel as a run-time library for the Ada 2005 GNAT compiler working on top of GNU/Linux. The second part of the work has been the design and implementation of the aforementioned services. For each of these new services different implementation alternatives were studied and their performance was evaluated. As a conclusion of the results obtained the new services can be implemented in an efficient way in an existing run-time system without requiring major changes. The implemented services have been made available to the clients of AdaCore, ARTIST2 partners, and to the community at large in the GPL version of the GNAT compiler.

Real-Time Java is an evolving technology. Over the last few years implementations have begun to appear and are now maturing. 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 year:

1. Community Building -The real-time Java user (academic and industrial) community was still fragmented. Although there had been a series of Real-Time Java Workshop (JTRES – Java Technology for Real-time and Embedded Systems), the attendance had been poor and the publications of the proceedings sporadic.

2. Maintaining Momentum - The main implementation approach of real-time Java, the Real-Time Specification for Java (RTSJ) needs further refinements; in particular some of the models needed more expressive power, and profiles for mission and safety critical applications need developing. Also, the application of the approach to modern architectures such as SMP and multicore platforms needs to be considered.

1.5 Comments From Year 2 Review

1.5.1 Reviewers’ Comments

“ACCEPTED

This task is aiming at studying RT languages and particularly ADA 2005 and its suitability in expressing RT constraints. The activity is starting and has not yet a lot of results to show which is normal. The document is very clear but would have slightly benefited to be a bit more concise. Interactions between partners are clearly appearing.

From telecom industry point of view, the reviewers would like to pin-point the importance in the future on real-time java which is a bit controversial at the moment.”

1.5.2 How These Have Been Addressed

The focus on Ada and Java continues. The work and attention given to Real-Time Java should help address some of the current controversial issues – although for Java many still remain.

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 Current Results

2.3.1 Technical Achievements

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

An initial set of utilities were presented at the International Real-Time Ada Workshop, IRTAW; they were subject to detailed review and have since been modified. Another workshop (an ARTIST2 event) will take place in Oct 2007 to discuss these modifications further. The utilities are based on a number of common real-time patterns that can be characterised under the following headings:

1) release conditions (event or time triggered),

2) fault model (deadline miss, WCET overrun or both; resumption or termination),

3) code structures for initialisation, normal behaviour (possible multi-moded) and error recovery,

4) Scheduling approach (Fixed priority or EDF),

5) Forms of protection and guarantee - servers of many different types. Servers are themselves being encoded for reuse.

The utilities so far produced have been verified by the use of a simulator. The availability of an implementation for Ada 2005 has only recently become available (see below). Once all utilities have been reviewed and tested on a real implementation they will be made available on an ARTIST2 web page.

One of the main ARTIST2 achievements of this year is its role in sponsoring and organising the 13th International Real-Time Ada Workshop. This took place in Woodstock, Vermont, and attracted participates from a number of European groups (ARTIST2 partners and non-partners) and the US. Full session summaries and workshop papers are available on the ARTIST2 web site, see http://www.artist-embedded.org/artist/IRTAW-13.html.

Implementation of Ada 2005 (Cantabria, York)

Some of the most exciting Ada 2005 innovations are those targeting the real-time community, providing capabilities that are neither addressed by other programming languages nor supported by most industrial execution platforms. The most important of these new services are: execution time clocks and timers, group execution time budgets, timing events, dynamic priorities for protected objects, immediate priority changes, and the new scheduling and task dispatching mechanisms. This audacious step forward reinforces Ada leadership for real-time programming, but needs to be endorsed by run-time systems that provide those new services.

As part of a joint effort with AdaCore, the GNAT run-time library has been adapted to run on top of MaRTE OS, which is a real-time operating system that can be configured as a POSIX-thread library for GNU/Linux. The first objective of this implementation is to provide a reference platform for GNU/Linux, fully compliant with Ada 2005, available for industrial, research, and teaching environments. Additionally, this platform can be used as a test bed before cross-development, providing a more user-friendly environment for testing and debugging.

Four features required to implement the new Ada 2005 real-time services have been fully implemented:

- Timing events are an effective and efficient mechanism to execute user-defined time-triggered procedures without the need to use a task or a delay statement. They have been implemented inside MaRTE OS, showing that they are much more efficient than the alternate use of timers and signal handlers, and therefore very useful also to POSIX applications.
- Execution time clocks. Monitoring execution time is important for many real-time systems, and Ada 2005 provides a standardized interface to obtain CPU consumption of every task in the system. Implementation has been straightforward as POSIX already supported this feature.
- Execution time timers are a timing mechanism that allows creating timers that are triggered when the execution time of a task reaches a given value, providing the means whereby action can be taken when this budget expires. The implementation has been based on an extension of timing events to also support execution time, and is therefore more efficient than the functionally equivalent POSIX timers and signal handlers.
- Dynamic priorities for protected objects. The priority ceiling of a protected object used for mutual exclusion and wait synchronization can be changed, and is useful for implementing “mode changes” or for utility libraries that include protected objects. The implementation has required introducing new facilities in MaRTE OS.

Low-level support for a fifth feature has been developed in MaRTE OS and now requires implementation inside the run-time system:

- Group execution time budgets allow creating and managing execution time budgets for groups of tasks. This capability has many practical applications in real-time systems in general, and therefore it is also interesting for real-time operating systems. An implementation of thread group budgets inside a POSIX real-operating system has been developed.

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

The Ravenscar profile for high-integrity systems is part of the Ada 2005. The UPM team has updated its Open Ravenscar kernel (ORK) for the LEON2 processor to the new standard, including some of the new real-time mechanisms:

- Execution-time clocks and timers.
- Group budgets.
- Timing events.

Execution-time timers and group budgets are not part of the Ravenscar profile, but they have been included in the enhanced version of ORK in order to experiment with the suitability of these mechanisms for implementing temporal isolation between critical applications running on the same computer platform. Preliminary results are encouraging, and the approach is being tested in a couple of pilot projects in the aerospace domain within the IST project ASSERT.

Plans for the future include implementing priority-band and heterogeneous scheduling (fixed priorities and EDF) in ORK.

Real-Time Java Community Building (UC3M, York)

ARTIST2 sponsored the 4th JTRES 2006 and York (Wellings) became Programme Chair and was charged with increasing the level of participation, and acquiring a permanent publication outlet. The result was that over 30 papers were submitted, signficantly more that at previous workshops. Each was reviewed by at least three members of the programme committee and 24 were accepted for presentation at the Workshop. The main technical programme consisted of papers on a variety of topics ranging from proposals for extensions and modifications to the RTSJ, through real-time garbage collection and other memory management issues to hardware support for embedded Java applications. A significant number of the papers addressed high-integrity real-time aspects. The work reported used a wide range of techniques from formal proof through model checking to results based on implementations and simulations. The Workshop was held at Conservatoire National des Arts et Métiers (**CNAM**), Paris, France during October 2006. It was attended by a healthy number of 68 people (34 from France, 8 from the UK, 7 from Spain, 5 from Italy, 4 from Germany, 2 from Denmark, 1 from Greece and Switzerland, 4 from the US and 1 from the Congo). There was an excellent mix of industrial and university participants (approximately 20 companies and 18 universities). The support from ARTIST2 allowed the registration fee to be kept to a minimum and this in turn resulted in the most successful JTRES workshop so far. The ACM Digital Library agreed to publish the proceedings. See http://www-users.cs.york.ac.uk/~andy/JTRES06 for details.

As a footnote, the improved attendance has carried forward into this year's workshop, where the number of papers submitted showed a small increase over last year's. See http://www.vmars.tuwien.ac.at/jtres2007/.

Real-Time Java Extensions (UC3M, York)

The European real-time community has significant experience in helping with the development of the Ada programming language and its use in real-time and safety critical applications. This experience has been used by York (Wellings) to impact on the direction that the RTSJ is taking. There are two Java Community Process Expert Groups that are currently looking at RTSJ issues. The first is JSR 282 that is looking at extending the RTSJ to provide more functionality. York is an active member of this community and participates in all its conference calls. The second is JSR 302 which is developing profiles for (initially) safety critical and (later) mission critical systems. The experience obtained from Europe in developing the Ada Ravenscar Profile and from the HIJA EU project have been instrumental in the approach adopted by the Expert Group, and again York is fully participating. The resulting initial specification issues which have been generated during year 3 of ARTIST2 are major achievements.

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. The work carried out at Universidad Carlos III de Madrid aimed at the construction of an architecture which offers platform independence; at the same time, it 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, and other contributions like the possibility of using asynchronous remote invocations or a centralized synchronization service which are useful in the development of many real-time systems. Initial programming interfaces, called DREQUIEMI, for this computation model have been already defined; they are 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) have been proposed to simplify the development of both, centralized and distributed real-time applications.

As a result of the European colaboration on real-time Java , a further EU project has recently been approved (JEOPARD) on the application of RTSJ technology to SMP and Multicore processors.

Memory management in the RTSJ (VERIMAG, York)

Dynamic memory management is a serious challenge for real-time embedded systems based on Java technology. Unlike the standard Java paradigm, garbage collection is rarely used in high-integrity real-time environments, since execution times and memory occupancy become difficult to predict and thus significantly complicates the implementation of real-time scheduling policies. Work internationally has addressed this problem from two perspectives. The first is to improve the efficacy of real-time garbage collection approaches, and the second is to support complementary memory management techniques such as region-based approaches. ARTIST2 partners have contributed in both these area.

York has been active in developing new garbage collection algorithms that integrate better with modern scheduling techniques (in this case dual-priority scheduling). Combining the advanges of fine-grained referencing counting with mark-and-sweep collection allows predictable responsive garbage collection to occur with well-defined timing properties. Integrating this with dual-priority scheduling allows gain-time and spare capacity to be effictively used.

The idea behind region-based memory management is to group objects of similar lifetimes: within a region, one can not deallocate any individual object, but must wait until the region can be destroyed as a whole. Regions are advocated by the RTSJ, which offers lexically scoped memory regions called ScopedMemory areas. This environment is appealing, as it guarantees constant-time memory operations, but it is very restrictive for the programmer: the size of the regions is fixed, and must be decided at programming time. Moreover, the RTSJ includes assignment rules that forbid an object in a short-lived region to be referenced by an older object. Programming for the RTSJ is thus very difficult, as it makes it impossible to reuse any old code (even the Standard Library has to be fully rewritten), and it forces the programmer to adopt new coding habits and to reason in a new paradigm quite different from Java.

In order to take advantage of the RTSJ without having to suffer from these drawback VERIMAG have proposed a technique to automatically transform standard Java code to a functionally equivalent scoped-memory managed code by analyzing the program to determine the lifetime of dynamically allocated objects. They have developed algorithms and tools to synthesize a memory organization that associates a memory region with each method in such a way the restrictions imposed by the RTSJ are fulfilled by construction. Thus, run-time checks can be safely eliminated to enhance performance. Moreover, they are also able to produce non-linear formulas which provide parametric upper-bounds of scope sizes in terms of program parameters.

In addition to this work, York has continued to investigate the general properties of coarse grained memory management, developing a model that allows the lifetime of coarse grain memory regions to be expressed. Called the Region Partitioning Model, it enforces a separation between memory concerns and program logic and thereby serves two purposes. Firstly, it allows an exploration into the repercussions of enforcing a scoped approach to coarse grain memory management as well as the RTSJ's implementation of this model. Secondly, by separating the memory management aspect from the application, it provides the basis for an environment offering low development and maintenance costs. The result is a model for developing RTSJ applications and a complementing extended RTSJ that addresses the prohibitive complexity of current development approaches while maintaining the performance and predictability advantages of the coarse grain model.

2.3.2 *Individual Publications Resulting from these Achievements*

York

A Burns and A. J. Wellings, “Concurrent and Real-Time Programming in Ada”. Cambridge University Press. July 2007

A. Burns and A.J. Wellings, Programming Execution-Time Servers in Ada 2005, Proceedings of the 27th IEEE Real-Time Systems Symposium, pp47-56, 2006.

Y. Chang and A. J. Wellings, “Hard Real-time Hybrid Garbage Collection with Low Memory Requirements”. In Proceedings of the 27th IEEE Real-Time Systems Symposium. pp. 77-86. December 2006.

O. M. Santos and A.J. Wellings, Formal analysis of aperiodic and sporadic real-time threads in the RTSJ, JTRES '06: Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems, pp10-19, 2006.

A. Borg and R.i Gao and N. Audsley, A co-design strategy for embedded Java applications based on a hardware interface with invocation semantics, JTRES '06: Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems, pp58-67, 2006.

Y. Chang and A.J. Wellings, Low Memory Overhead Real-Time Garbage Collection for Java, JTRES '06: Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems, 2006.

A. Zerzelidis and A.J. Wellings, Model-based Verification of a Framework for Flexible Scheduling in the Real-Time Specification for Java, Proceedings of the 4th international workshop on Java technologies for real-time and embedded systems, pp20-29, 2006.

A. Zerzelidis, A. Burns and A.J. Wellings, Correcting the EDF protocol in Ada 2005. Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

A. Burns and A.J. Wellings, Programming Execution-Time Servers in Ada 2005, Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

A.J. Wellings and A. Burns, A Framework for Real-Time Utilities for Ada 2005, Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

A.J. Wellings and A. Burns, Integrating OOP and Tasking – The missing requeue, Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

A.J. Wellings and A. Burns, Beyond Ada 2005: Allocating Tasks to Processors in SMP Systems, Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

M. Ward and N.C. Audsley, Suggestions for Stream Based Parallel Systems in Ada, Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

A. Burns and A.J. Wellings, Programming Execution-Time Servers and Supporting EDF Scheduling in Ada 2005, Handbook of Real-Time and Embedded Systems, Chapman and Hall/CRC, Eds:Insup Lee, Joseph Y-T. Leung and Sang H. Son, pp13-1--13-20, 2007.

A.J. Wellings and A. Burns, Real-Time Java, Handbook of Real-Time and Embedded Systems, Chapman and Hall/CRC, Eds: Insup Lee, Joseph Y-T. Leung and Sang H. Son, pp12-1--12-17, 2007.

A.J. Wellings and A. Burns, Real-Time Utilities for Ada 2005, Reliable Software Technologies - Ada Europe 2007, Lecture Notes in Computer Science, Vol 4498, pp1-14, 2007.

Y. Chu and A. Burns, Supporting Deliberative Real-Time AI Systems: A Fixed Priority Scheduling Approach, Proceedings of ECRTS07, pp259-268, 2007.

A. Borg and A. J. Wellings, ”Scoped, Coarse-grain Memory Management and the RTSJ Scoped Memory Model in the Development of Real-time Applications”. International Journal of Embedded Systems, Volume 2, Number 3/4, pp. 166-183. 2006.

D. Kazakov and I. Bate, “Towards New Methods for Developing Real-Time Systems: Automatically Deriving Loop Bounds Using Machine Learning”. In Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation, Sept. 2006.

Universidad de Cantabria

M. Aldea-Rivas, J. F. Ruiz. Implementation of new Ada 2005 real-time services in MaRTE OS and GNAT 12th Ada-Europe International Conference on Reliable Software Technologies, Geneva, Switzerland, June 25-29, 2007, in Lecture Notes in Computer Science , Vol. 4498.

Mario Aldea Rivas and Michael González Harbour. Operating System Support for Execution Time Budgets for Tread Groups, Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

Porto

Nogueira, L., Pinho, L., "Capacity Sharing and Stealing in Dynamic Server-based Real-Time Systems", 21st IEEE International Parallel & Distributed Processing Symposium, Workshop on Parallel and Distributed Real-Time Systems, Long Beach, EUA, March 2007.

Nogueira, L., Pinho, L., "Handling Shared Resources and Precedence Constraints in Open Systems", WiP Session of the 19th Euromicro Conference on Real-Time Systems (ECRTS’07), Pisa, Italy, July 2007.

Universidad Politécnica de Madrid

S. Urueña, J. Pulido, J. Redondo, J. Zamorano. Implementing the new Ada 2005 real-time features on a bare board kernel. Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

S. Urueña, J. Zamorano. Building High-Integrity Distributed Systems with Ravenscar Restrictions. Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

J. Pulido, J.A. de la Puente, M. Bordin, T. Vardanega, J. Hugues. Ada 2005 Code Patterns for Metamodel-Based Code. Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear).

José A. Pulido, Santiago Urueña, Juan Zamorano, Juan A. de la Puente. Handling Temporal Faults in Ada 2005. In Nabil Abdennadher and Fabrice Kordon (eds.), Reliable Software Technologies — Ada-Europe 2007, LNCS 4498, pp. 15–28. Springer-Verlag, 2007.

Juan Zamorano, Juan A. de la Puente, Jérôme Hugues, Tullio Vardanega. Run-time mechanisms for property preservation in high-integrity real-time systems. OSPERT 2007 — Workshop on Operating Systems Platforms for Embedded Real-Time applications. Pisa, Italy, July 2007.

Universidad Carlos III de Madrid.

Pablo Basanta-Val, Marisol García-Valls, Iria Estévez-Ayres and Carlos Delgado-Kloos. ExtendedPortal: Violating the assignment rule and enforcing the single parent one. In 4th International Workshop on Java Technologies for Real-Time and Embedded Systems JTRES06, pp30-37, Paris, France. ISBN 1-59593-544-4, Oct 2006.

Basanta-Val, P., Almeida, L., Garcia-Valls, M., and Estevez-Ayres, I. 2007. Towards a Synchronous Scheduling Service on Top of a Unicast Distributed Real-Time Java. In Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium. RTAS. IEEE Computer Society, Washington, DC, pp123-132, 2007.

VERIMAG

G. Salagnac, Ch. Rippert, S. Yovine. Semi-automatic Region-Based Memory Management for Real-Time Java Embedded Systems. In Proceedings of 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'07), August 2007. IEEE Computer Society Press.

2.3.3 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 others sites take place and interactions via email allows 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) fulfiles 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.

2.3.4 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.

The session reports of the 13th IRTAW were the result of joint work at and during the workshop. The six reports will appear both in the Proceedings of the 13th IRTAW, Volume XXVII Issue 2, ACM AdaLetters, August 2007 (still to appear), and in future issues of the Ada User Journal.

2.3.5 Keynotes, Workshops, Tutorials

13th International Real-Time Ada Workshop - IRTAW
Woodstock, Vermont, USA, 17th-19th April 2007.

Full session summaries and workshop papers are available on the ARTIST2 web site, see http://www.artist-embedded.org/artist/IRTAW-13.html .

4th Java Technology for Real-Time and Embedded Systems - JTRES
Conservatoire National des Arts et Métiers (CNAM), Paris, France, Oct 2007.

See http://www-users.cs.york.ac.uk/~andy/JTRES06 for details of the event and its papers.

3. Future Work and Evolution

3.1 Problem to be Tackled over the next 12 months (Sept 2007 – Aug 2008)

During the third year attention has been focused on the Ada programming language because it has recently completed a revision (Ada 2005) that incorporates a number of features that are especially useful to real-time systems engineers. This work will continue during the next year as patterns of use are developed, analysed and published. Work will also continue on the Real-Time Specification for Java (RTSJ) as it is also currently been updated. A number of the facilities recently incorporated in Ada 2005 are also being considered for inclusion in the RTSJ. There may also be future modifications to the provisions of POSIX that will have an influence on the use of the POSIX API in languages such as C and C++.

To broaden the focus of this activity there will also be an effect during this year to produce a white paper that will summarise language development efforts over a wider constituency of languages. To this end ARTIST2 is sponsoring, and helping to organise (an ARTIST member, Burns, is co-chair) SYNCHRON'07 the International Open Workshop on Synchronous Programming: http://www.gdi.wiai.uni-bamberg.de/Synchron2007/ in Bamberg, Germany during November 2007.

Work will also continue on the evaluation of Ada 2005 and the RTSJ real-time capabilities for EDF-based server scheduling (sharing and stealing servers), and the use of proof-carrying code with Ravenscar.

One of the challenges for language designers (and operating system developers) is to respond to the difficulties arising from the use of increasingly sophisticated execution platforms. These platforms have many cores and busses, networks and dynamic hardware such as FPGA. It is far from clear what the right programming abstractions are for these platforms. Even the relatively simple task of dealing with the static and dynamic placement of parallel code on SMPs is not adequately addressed in modern programming languages. The impact of platform design on language development will continue to be a focus of research that will be monitored by this activity.

3.2 Current and Future Milestones

- Year 3 Milestone:
    - Organise and participate in the 13th IRTAW. (*This was organised and the event was very successful).*
    - Publish via a web site an initial set of patterns (repository) for use by Ada 2005 (*This was delayed by the availability of the Ada 2005 compiler, however patterns have been developed and once tested will be placed on the Web site).*
- **Milestones for final year**
    - **Publish via a web site a set of patterns (repository) for use by Ada 2005 application programmers. This is left over from Year 3 (see discussion above).**
    - **Produce a white paper linking all real-time language work within ARTIST2 partners (including reference to external research effort where appropriate). The aim of this paper will be to compare the various language development**

efforts occurring in different research and industrial sectors in order to ease the cross-over of key notions and abstractions.

- **Extend the Ada 2005 repository to include a number of scheduling examples and server constructs.**
- **Complete the implementation of all the Ada 2005 real-time features, in particular the support for EDF scheduling.**

3.3 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.
- Joint work on a white paper on programming languages for real-time systems.
- Involvement in the planning of workshops.

3.4 Main Funding

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

- 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.

There are currently a number of new EU projects under negotiation that should start during the final year of ARTIST2.

4. Internal Reviewers for this Deliverable

Michael González Harbour (Universidad de Cantabria)

Peter Puschner (Technische Universitaet Wien)

Andy Wellings (University of York)