ARTIST 2

Network of Excellence

IST-004527 ARTIST2:
Embedded Systems Design

Activity Progress Report for Year 2

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/%7Eburns/

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 RTSJ (Real-Time Specification for Java), C (with POSIX) and SCOOP (a concurrent real-time extension to Eiffel).

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 ARTIST to influence the development of both engineering languages such as Ada and RTSJ (Real-Time Specification for Java) and research languages such as SCOOP (a collection of possible extensions to Eiffel).

Main focus this year has been on the new Ada 2005 standard and the planning of a number of activities for the coming year. During this reporting period, one internal ARTIST meeting/workshop on Ada 2005 has taken place and a public workshop on SCOOP. The Ada meeting started to define some reusable patterns for flexible applications. The SCOOP workshop identified a number of topic requiring further study before a language like Eiffel, with it useful formal underpinning, can be used in the real-time domain.

1.1   ARTIST2 Participants: Expertise and Roles

Professor Alan Burns – University of York (UK)

> *One of the language designers for Ada and 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 evaluates 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 evaluates of, the proposed extensions to Ada, Java and POSIX.*

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.*

1.2   Affiliated Participants: Expertise and Roles

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

> *Worked on 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 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.*

Sergio Yovin – VERIMAG

> *work on real-time Java*

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.

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 version; 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 define set of primitives for undertaking fixed priority (i.e. essentially static) scheduling for non-adaptive applications. The research community, including members of the ARTIST community, has been involved in defining new language feature 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 ARTIST members).

Ada 2005 is now defined and is undergoing international standardisation. It is appropriate therefore to assess its 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 event, 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. ARTIST provided the framework for this broad set of activities to compare outcomes and to influence each other's research.

1.5   Problem Tackled in Year2

Effort has been 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 and an international open workshop on SCOOP. For both of these, see descriptions below.

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.   Summary of Activity Progress

2.1   Previous Work

Over the last two decade or more, a considerable volume of literature has been produced which addresses the issues and challenges of scheduling and adaptive applications. The focus of much of this research has been on how to effectively support a collection of distinct applications each with a mixture of periodic and non-periodic, and hard and soft, activities. This work is seen as an enabling technology for a wide range of applications from multi-media to robust control. Notwithstanding the quality of individual contributions, it is unfortunately true that system implementers have not, in general, taken up these results. There are a number of reasons for this, including the normal inertia associated with technical change, but we would highlight the following:

- Inconsistency in the literature - no single technique has emerged as being widely applicable; indeed there is not even any consensus over the right simulation models to use for evaluation.

- Limitations of the scheduling models - too many models have unrealistic assumptions such as ignoring run-time overheads or assuming that the WCET is known for all tasks (or not recognizing that actual execution times can be much smaller than WCET values).

- Difficulty of implementation - often the scheduling scheme will require operating system primitives that are not available on any commercial platform.

- Lack of computational model - the scheduling results are not tied back to realistic models that applications can use to construct systems.

- Lack of design patterns that applications programmers can adopt and adapt.

Many of these problems can be alleviated by having the right models and abstractions in the implementation programming language, and by the development of pattern that make use of these language features.

Over the last few years, the appropriate abstractions that are needed to support the development of flexible real-time systems have been the subject of much investigation. Working in conjunction with the Series of International Workshop on Real-Time Ada Issues – IRTAW (Proceeding published in Ada Letters, Vol XXI, March 2001, Vol XXII, December 2002 and Vol XXIII, December 2003.), a set of mechanisms have been defined that allow a wide range of flexible real-time systems to be implemented. These have now been incorporated into the Ada 2005 draft standard. Many ARTIST members have contributed to this work and hence it is an effective use of the ARTIST framework to continue these interactions.

2.2   Current Results

2.2.1   Technical Achievements / Outcomes / Difficulties encountered

Work on Ada 2005

A number of sites have started to evaluate the new features available in Ada 2005. A useful meeting was held in March (see description below).

The university of York has begun to build a library of reusable facilities to ease the programming of flexible real-time systems. This has concentrated on the programming of standard patterns such as periodic tasks (with deadline and CPU overrun detection), and sporadic tasks (with deadline, CPU time and minimal separation violation detection).

The University of Cantabria has been working on the integration of the real-time services built in the MaRTE OS kernel with the GNAT Ada compiler system, with the goal of generating a platform that fully supports the new real-time features defined in the Ada 2005 standard. This platform will be used by the other partners involved in this activity to integrate further research in the area of real-time languages, and will be made available to industry and the Ada community in general as free software.

The particular activities carried out in this period at the University of Cantabria have been the addition of the timed event service to the MaRTE OS kernel, the integration of this facility with the GNAT run-time system, and the study of the changes required in the GNAT compilation system for supporting Ada 2005 real-time services. In addition, a development environment has been setup for the integration of the compiler run-time system with the kernel, and the entire Ada official test suite has been run in order to validate the new platform.

Work on other languages

Ada has in many ways lead the way in attempting to bring into language design the abstractions and idea that have been developed within the real-time research community. Other languages have are also moving in that direction. The C language has been closely linked to POSIX and ARTIST is participating in POSIX standardisation work (although this is not a very active area at the moment). Java, through its RTSJ (real-time specification for Java), has many of the flexibility features desirable in an implementation language – again ARTIST is fully involved in the definition work surround RTSJ. ARTIST is therefore sponsoring and co-organising JTRES (see future work). The work that has been undertaken on Java and Ada has also been used to influence the develop of concurrent and real-time versions of Eiffel (SCOOP). Again ARTIST sponsored a workshop on this language topic.

Other work has been focused on research languages. AbsInt participates in the ST-project Embounded (www.embounded.org). Within Embounded a new functional programming language (Hume) for real-time applications is under consideration. The aims of the EmBounded Project are to identify, to quantify and to certify resource-bounded code in Hume, a domain-specific high-level programming language for real-time embedded systems. Using formal models of resource consumption as a basis, the project will develop static analyses for time and space consumption and assess these against realistic applications for embedded systems.

In almost all real-time systems, the chosen implementation language, the software synthesis method and/or the coding guidelines have a strong influence on the analyzability of real-time systems. Disciplined code as synthesized from specification languages like SCADE, ASCET-SD and others allow the determination of safe and highly precise execution-time bounds. Undisciplined code and the use of many dynamic languages features, e.g. dynamic method dispatch or dynamically allocated heap data structures, lead to high degrees of overestimation. It follows that there is a link between the work of this activity and that undertaken on WCET elsewhere in ARTIST2. This link will continue to be explored.

2.2.2 Publications Resulting from these Achievements

None in this reporting period, *(starting March 2006).*

2.2.3 Keynotes, Workshops, Tutorials

Keynotes

Andy Wellings from the University of York gave the invited keynote presentation at CORDIE (see below). The title of this talk was: "RECOOP: Real-Time SCOOP for Eiffel".

Workshop : Ada 2005

York, 22/23rd March, 2006

A two day workshop/meeting for this activity was held in York with 12 participants from York, Cantabria, Madrid, Porto and Vienna. The agenda for the meeting was as follows:

1. Planning for next IRTAW (International Real-Time Ada Workshop).

2. Review of Ada 2005 run-time implementation effort by group in Cantabria

3. Planning for next ARTIST2 meeting (in Spain during September)

4. How to get the Ada message out to the rest of ARTIST and Europe!

5. Review of new Ada 2005 standard – in particular last minutes changes

6. Discussion on the programming of servers

7. New programming paradigms that Ada 2005 support/enables

8. Use of Ada in education programmes.

9. Defining Milestones for this activity

The final set of changes to the Ada 2005 standard was outlined by Burns (a member of the Ada Standardisation working group of ISO). Overall the new standard has added considerable expressive power to Ada in the area of real-time systems. The new language now has (Java-like) interfaces, including synchronised, task and protected, new dispatching policies including round robin and EDF (earliest deadline first), budget control over tasks and groups of tasks, and a new abstraction of a timing event – this is similar to an interrupt but is fired by the passage of time.

Although the new language design has introduced these addition features it is still necessary for run-time systems to be developed that can make the abstractions available. It will be necessary to evaluate the features both in terms of there inherent efficiency and in terms of their ease of use (how well can the feature be used together). Harbour discussed the work at Cantabria to update their MaRTE Ada kernel.

A large part of the meeting was focused on a series of presentations and discussions on the programming of servers and the right means of capturing the different sets of semantics that are possible. Improvements to the proposals were made – these will be published and made available via a web-based repository (probably housed at York). It is through the development, verifications and publication of these patterns and their interfaces that the work of this ARTIST activity will have most effect on the end users. It is envisaged that the coming year will see the first set of such patters to be made available (via a web site incorporated into the ARTIST2 public web service).

The remaining time of the meeting was spent planning future events, in particular the next IRTAW which will be held in US in March next year – with sponsorship by ARTIST and a number of ARTIST members planning the event, including de la Puente who has the role of PC chair.

Workshop : CORDIE   *York, 4/5 July, 2006*

http://www.artist-embedded.org/FP6/ARTIST2Events/Events/Cordie06/

ARTIST2 personnel and financial support were provided for the first CORDIE workshop. Participants were able to bring to the discussions the needs of the flexible real-time community. Publications are likely to result in the coming year.

The First Symposium on Concurrency, Real-Time, and Distribution in Eiffel-like Languages (CORDIE) took place on 4/5 July 2006 in York. The focus of the conference was on adding concurrency and real-time features to object-oriented languages that support contract-based methods (e.g., Eiffel/Spec# pre- and postconditions of methods), and on the tradeoffs that must be considered when attempting to make such additions.

The symposium featured two full days of invited presentations and presentations of fully refereed technical papers, discussing issues from detailed semantics of concurrency (e.g., when to lock objects, when to release objects, scheduling), to real-time issues (e.g., adding timeouts, deadlines, and mode changes), to implementations, to verification (e.g., reasoning using fair transition systems), to distribution issues (e.g., distributing objects across processors). The presentations sparked lively discussion and debate, and led to substantial discussion and technical work outside of the formal sessions.

The invited speakers were:
- Prof. Andy Wellings, University of York: "RECOOP: Real-Time SCOOP for Eiffel"
- Prof. Antonia Lopes, University of Lisbon: "Community"
- Prof. Jonathan Ostroff, York University, Canada: "Tutorial: An Introduction to SCOOP"
- Prof. Bertrand Meyer, ETH Zurich: "Closing Address: One SCOOP or More?"

The international programme committee (see http://www.artist-embedded.org/FP6/ARTIST2Events/Events/Cordie06/Organisation.php) reviewed all submissions and recommend acceptance of 9 papers, out of 11 submissions. The accepted papers were:

- Piotr Nienaltowski, Bertrand Meyer: "Contracts for Concurrency"

- Jonathan Ostroff, Faraz Torshizi, Hai Feng Huang: "Verifying Properties Beyond Contracts of SCOOP Programs"

- Gordon Jones, Emmanuel Bouyer: "Why We Didn't Use SCOOP"

- Phil Brooke, Richard Paige: "A Critique of SCOOP"

- Volkan Arslan, Bertrand Meyer: "Asynchronous Exceptions in Concurrent Object Oriented Programming"

- Piotr Nienaltowksi: "Flexible Locking in SCOOP"

- Miguel Oliveira e Silva: "Automatic Realizations of Statically Safe Intra-Object Synchronization Schemes in MP-Eiffel"

- Gordon Jones, Emmanuel Bouyer: "Reliable Distributed Eiffel Components"

- Phil Brooke, Richard Paige: "An Alternative Model of Concurrency for Eiffel"

The symposium was attended by 23 participants, mostly from the EU, with others from Canada.

Feedback from the participants was overwhelmingly positive, with all noting the enjoyable technical discussions, the constructive debate, and the quality of the invited speakers, who were chosen so as to bring different perspectives to the community of users of Eiffel-like languages. The participants noted a substantial desire to organise a follow-up meeting, and this will likely be proposed as a workshop for ECOOP'07, which would be held at TU-Berlin in July 2007.

3.   Future Work and Evolution

3.1   Problem to be Tackled over the next 18 months (Sept 2006 – Feb 2008)

A number of workshops are planned for the coming period, including the next International Real-Time Ada Workshop (IRTAW13) which will take place in the US (Boston area) in March 2007. This will be sponsored by ARTIST and involve a number of ARTIST participants (see earlier discussion). ARTIST is also co-organising JTRES – 4th Workshop on Java Technology for Real-time and Embedded Systems. This is scheduled for 11-13 Oct 2006 in Paris. See: http://www.artist-embedded.org/FP6/ARTIST2Events/Events/JTRES-2006/

Work will continue on the evaluation of Ada 2005 including the use of new Ada 2005 real-time capabilities for EDF-based server scheduling (including capacity sharing and capacity stealing servers), and the use of proof-carrying code in Embedded Systems. Much of this work will be reported to IRTAW13. A further ARTIST meeting to look at implementation issue will occur in Spain in October 2006.

Both of the above events will be used to define and extend the pattern of use that will allow these languages to be used in the programming of effective, adaptive and flexible real-time systems.

This activity will also continue to act as a coordinator for a wide range of language work going on within Europe.

3.2   Current and Future Milestones

The milestones for this activity (as defined during our first activity meeting) are as follows. Note activity started in March 2006, so Year 3 milestones are after 18 months of activity

- Year 3

    o Organise and participate in the 13th IRTAW.

    o Publish via a web site an initial set of patterns (repository) for use by Ada 2005 application programmers.

    o Publish via a web site the outcomes of the JTRES workshop.

- Year 4

    o Produce a white paper linking all real-time language work within ARTIST partners (including reference to external research effort where appropriate).

    o Extend the repository

3.3   Indicators for Integration

Joint involvement in planning and participating in workshops. Joint commitment to construct a web-based repository of Ada pattern for real-time algorithms and structures.

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.

- THREAD Spanish project, in which the following ARTIST2 partners are involved: Technical University of Madrid, University of Cantabria,Technical University of Valencia.