European Cooperation in the field of Scientific and Technical Research - COST -

Brussels, 15 May 2014

COST 042/14

MEMORANDUM OF UNDERSTANDING

Subject :	Memorandum of Understanding for the implementation of a European Concerted Research Action designated as COST Action IC1402: Runtime Verification beyond Monitoring (ARVI)

Delegations will find attached the Memorandum of Understanding for COST Action IC1402 as approved by the COST Committee of Senior Officials (CSO) at its 190th meeting on 14 May 2014.

MEMORANDUM OF UNDERSTANDING
For the implementation of a European Concerted Research Action designated as

COST Action IC1402
RUNTIME VERIFICATION BEYOND MONITORING (ARVI)

The Parties to this Memorandum of Understanding, declaring their common intention to participate in the concerted Action referred to above and described in the technical Annex to the Memorandum, have reached the following understanding:

1. The Action will be carried out in accordance with the provisions of document COST 4114/13 “COST Action Management” and document COST 4112/13 “Rules for Participation in and Implementation of COST Activities” , or in any new document amending or replacing them, the contents of which the Parties are fully aware of.

2. The main objective of the Action is to consolidate a network of runtime verification experts and practitioners in application domains, so that they jointly find new principles for reliable system engineering using monitoring as a building block.

3. The economic dimension of the activities carried out under the Action has been estimated, on the basis of information available during the planning of the Action, at EUR 52 million in 2014 prices.

4. The Memorandum of Understanding will take effect on being accepted by at least five Parties.

5. The Memorandum of Understanding will remain in force for a period of 4 years, calculated from the date of the first meeting of the Management Committee, unless the duration of the Action is modified according to the provisions of section *2. Changes to a COST Action* in the document COST 4114/13.

A. ABSTRACT AND KEYWORDS

Runtime verification (RV) is a computing analysis paradigm based on observing a system at runtime to check its expected behavior. RV has emerged in recent years as a practical application of formal verification, and a less ad-hoc approach to conventional testing by building monitors from formal specifications.

There is a great potential applicability of RV beyond software reliability, if one allows monitors to interact back with the observed system, and generalizes to new domains beyond computers programs (like hardware, devices, cloud computing and even human centric systems). Given the European leadership in computer based industries, novel applications of RV to these areas can have an enormous impact in terms of the new class of designs enabled and their reliability and cost effectiveness.

This proposal aims to build expertise by putting together active researchers in different aspects of runtime verification, and meeting with experts from potential application disciplines. The main goal is to overcome the fragmentation of RV research by (1) the design of common input formats for tool cooperation and comparison; (2) the evaluation of different tools, building a growing sets benchmarks and running tool competitions; and (3) by designing a road-map and grand challenges extracted from application domains.

Keywords: Runtime Verification, Monitoring for System Design, Software Reliability, System Reliability, Monitor reflection and execution replay

B. BACKGROUND

B.1 General background

Many system failures through history have exposed the limitations of existing engineering methodologies and encouraged the emergence of formal methods. Ideally, one would like to validate a program or computing system prior to its execution. However, current static validation methods, such as model-checking, suffer from practical limitations preventing their use in real large-scale applications. For instance, those techniques are often bound to the design stage of a system and suffer from the state-explosion problem (the unfeasability to explore all system states statically), or cannot handle many interesting behavioral properties. Thus, as of today many verification tasks can only realistically be undertaken by complementary *dynamic analysis* methods. The discipline of formal dynamic analysis is called runtime verification and studies how to detect

and ensure, at execution time, that a system meets a desirable behavior.
Even though research on runtime verification has flourished in the last decade (see the conference series at http://runtime-verification.org), it is geographically dispersed, with different research groups in Europe (and elsewhere) starting from different background and motivated by particular application objectives. We are at a point when this applied field of formal methods and rigorous software engineering can greatly benefit from a networking effort both within the community and with potential areas of application, including finances, medical devices, and embedded, cloud and distributed systems.
Several of the Action partners collaborated in COST Action IC0901 in many occasions discussing issues related to runtime verification, realizing clearly that the research field of RV was fragmented and could greatly benefit with the help of a networking effort. In this COST Action we put together the European active experts on RV and foster interaction with domain experts to overcome this fragmentation.

B.2 Current state of knowledge

Runtime verification has evolved in recent years. In most cases, these developments have focussed to a specific problem area, naturally missing the necessary generality to be compared, reused and applied to other domains, and it is the aim of this Action to overcome this.
RV has gained popularity both in formal verification and software engineering, because in most of the cases it reveals to be more practical than static exhaustive verification techniques and more versatile than classical software testing. RV is generally used when the system model is too big to be handled by model checking or when the system model is not completely available, but only as a black-box where only the inputs and outputs are observable. Various runtime verification tools have been built and used for different purposes---EAGLE, jUnitRV, LOLA, Larva, polyLarva, JavaMOP and Java-MaC just to mention a few. They differ in various ways---like the logics they monitor, technologies to which they can be applied, level of integration of the monitors with the system, etc. Comparing different systems requires to identify comparison dimensions, like expressivity, efficiency, applicability, system availability, online or offline monitoring, etc. The diversity and lack of benchmarks for the objective comparison of different toolsets is now being recognized, and one can see the first attempts at this through the establishing of a first runtime verification competition to be held as part of Runtime Verification 2014 (see http://rv2014.imag.fr/monitoring-competition). The recent tool SMock provides a first attempt to build a test platform for runtime verification tools, but there is more work to provide a general infrastructure for comparison, which

can only be better accomplished through networking.

Some RV tools have explored the problem of computing aggregations, quantitative and statistical properties, richer than what static formal methods and software testing typically handle. The approaches proposed differ in their theoretical foundations (algebraic automata, streams, regular-like expressions, rules, first-order temporal logics, etc), and are implemented in tools like LarvaStat, LOLA and EAGLE. Monitoring distributed systems and properly decomposing monitors has only been studied in preliminary theoretical work.

Instrumentation is perceived as one of the most important issues in runtime verification today. More often, directly instrumenting a system can create a significant overhead in the monitoring process. Some techniques have been recently proposed, including sampling the monitored system and using static analysis to better instrument the system, but these are still very active lines of research. Again, proper techniques and tool cooperation and comparison in all these aspects is missing.

RV has been successfully used to compensate the intrinsic incompleteness of testing coverage by integrating verification oracles in the code, allowing for reparatory actions to be taken in case of system failure after deployment. However, most of the work on combining testing and RV separate the stages of the software life-cycle resulting in large duplication. The other direction, that is, using testing infrastructure to improve RV is mostly unexplored. Since testing is well-established in industry using testing to support monitoring can be particularly effective in encouraging industry pickup, accelerating the adoption of runtime verification into software practices.

Most advanced variations and extension of runtime verification, like reflection (the power of a monitor to influence the monitored system tool), robustness with respect to the precision of the observations, exploiting the trade-off between observation precision and monitoring overhead, online replay based on error detection are largely unexplored and have a great potential applicability.

B.3 Reasons for the Action

Reasons for the Action are coordinating runtime verification research efforts, to allow the comparison and collaboration of approaches, methods and tools, and to put together the whole community of core experts with practitioners from application domains.

Runtime verification provides additional reliability on top of classical software testing. Substantial benefit is expected in particular in application domains with safety-critical systems or with a high demand for flexibly connecting interacting systems. High quality standards in challenging

applications constitute the basis for a European competitive advantage.
Even though research on runtime verification has flourished in the last decade, this research is fragmented, not only geographically dispersed with different research groups mostly in Europe and the US, but they also started from different backgrounds and are motivated by particular application objectives. These objectives are often guided by the application domain of practitioners that each research group collaborates with. The potential of a coordinated network on runtime verification fostering cross fertilization is much higher than that of the individual groups.
The researchers and practitioners are at a point when this applied field of formal methods and rigorous software engineering can greatly benefit from a networking effort both within the community and with areas of application, especially in the medical and legal domain. The Action is aimed at scientific/technological advance and European economical/societal needs.

B.4 Complementarity with other research programmes

The research topics of the Action are currently supported by the following independent national programs and EU grants.
The ongoing OR.NET project ("Sichere dynamische Vernetzung in Operationssaal und Klinik", 2012--2015), funded by the German federal ministry of education and research (BMBF), focuses on the dynamic interconnection of medical devices. RV is used to increase the safety of the resulting medical devices.
The ongoing EU FP7 project ENVISAGE ("Engineering Virtualized Services", 2013--2016) develops a practical open-source framework for model-based development of virtualized services. RV is used to monitor and adapt deployed services with regard to their (formalized) service level agreements.
The ongoing project STRONGSOFT ("Sound Technologies for Reliable, Open, New Generation Software"), funded by the Spanish Ministry of Economy and Competitiveness, attempts to build the foundation to a new generation of reliable software. This project includes one activity on establishing foundations for RV of heterogenous and concurrent systems.
The ongoing project StaRVOOrS ("Unified Static and Runtime Verification of Object-Oriented Software", 2013--2016), funded by The Swedish Research Council, aims at developing a formal language to specify properties to be checked statically and at runtime. The core idea is to automatically prove statically as much as possible, automatically obtaining a runtime monitor for those proof obligations that cannot be proved statically.

The recent EU FP7 project PINCETTE ("Validating Changes and Upgrades in Networked Software", 2010 - 2013) addresses the problem of high cost of changes by introducing an automated framework and methodology, and a mix of static and run-time analysis technologies to identify the impact of changes that derive from intra-component changes (due to error fixing and functionality enhancement) and from component replacement within a single product and a product family. This methodology improves the reliability of networked software by implementing an innovative solution for the automatic detection, localization, and repair of program bugs.

The ongoing project "Secrecy and Information Flow in Shared Document Bases", funded by the German Research Foundation as part of the priority programme "RS3: reliably secure software systems", develops static and dynamic techniques for the analysis of secrecy and information flow in applications where multiple users execute a workflow while sharing a common document base.

The transregional research center "AVACS: Automatic Verification And Analysis of Complex Systems", funded by the German Research Foundation, addresses the rigorous mathematical verification and analysis of models and realizations of complex safety-critical computerized systems, such as aircraft, trains, cars, or networks formed by these, whose failure can endanger human life.

Of the ongoing COST Actions, the closest one to this proposal is IC1201 (BETTY) on behavioral types for reliable large-scale systems. Both aim at methods and tools for enforcement of software contracts and overlap in some of fundamental machinery used (eg. automata/languages). In BETTY however the focus is on static enforcement of lightweight contracts specifically addressing interaction while in this Action we combine static techniques with dynamic checking to enforce stronger contracts going beyond types. The Actions are thus complementary, yet it is planned to have a joint meeting for exchanging ideas.

The activities listed above will serve as sources of cases studies, tools and motivations. This Action will include and collaborate with researchers involved in these activities. However, no prior activity by itself proposed to develop a unifying framework for runtime verification that allows comparing tools and methods, and that aims at bringing together large communities of practitioners with the whole body of runtime verification experts.

C. OBJECTIVES AND BENEFITS

C.1 Aim

The main objective of the Action is to consolidate a network of runtime verification experts and to build collaboration between these experts and practitioners in application domains, especially medical devices and legal bodies, so that they jointly find new principles for reliable system engineering using monitoring as a building block.

C.2 Objectives

To achieve the overall aim, the main objectives are:

- the development of a common infrastructure that enables the development of a collection of runtime verification problems and benchmarks for the comparison of algorithms and tools, and to increase their collaboration
- the development and sharing of current challenges in runtime verification and monitoring
- the development of an interaction between the runtime verification community of experts at large with practitioners from application domains that could benefit from this technology, and influence its developments
- education of young researchers and potential users of monitoring technologies
- coordination of European research on monitoring, runtime verification and its applications

C.3 How networking within the Action will yield the objectives?

The objectives will be achieved through:

- Regular Working Group meetings, twice per year.
- a series of workshops specifically dedicated to challenges and especially applications for runtime verification, where invited industrial practitioners will interact with the RV community of experts including information and problem exchange, once per year
- Short-Term Scientific Missions, exchanging PhD students to materialize research collaborations
- publications of results in leading computer science conferences and journals

- development of case studies and grand challenges for monitoring in new application domains
- maintenance of an online reference in the area of runtime verification and monitoring
- Summer Schools organized in the last two years of the Action

C.4 Potential impact of the Action

This Action will coordinate the European efforts in the field of runtime verification and applications. The main benefits of this COST Action are inherent to the coordination of otherwise fragmented common research activities: better understanding of problems and solutions by comparison and cooperation, better tools by sharing implementations, reuse tool components and inputs. Also, a creation of a network that connects runtime verification researchers with industrial experts will allow to identify new goals and research directions and to attack them cooperatively, and to find and to influence new applications in these domains.

In addition to these benefits, concrete outcomes of the Action will include (1) a taxonomy of problems and techniques, and a taxonomy of existing tools, (2) a common family of input languages for describing problems and solutions, (3) a collection of benchmarks that allows to compare the different tools, (4) a set of challenges for the applicability of runtime verification to important areas of application.

The adoption of the runtime verification techniques developed in this COST Action have the impact of improving and enabling new designs of higher-quality systems, more robust and less prone to errors occurring at runtime, and more suitable for automatic repair. Additionally, due to the ubiquity of computer based systems and computer controlled systems in modern life and the applicability of monitoring beyond computer systems, this impact will be beneficial for the European society at large.

C.5 Target groups/end users

Apart from the existing research community on RV, we also target hardware and software designers and developers. They will benefit from the Action by having more tools for specifying and implementing monitoring and adaptation techniques, independent from their particular programming environment, at their disposal.

A further target are educators in software engineering, which will find a consolidated methodology

and inventory of RV tools and techniques that will allow them to inform the participants in their courses about domain-independent means of solving particular problems.
Through the involved industry partners, we will specifically target areas where contracts that describe contracted/provided services are used, and the medical domain. This will complement academic activities and give a concrete context in which RV tools and techniques can be employed and evaluated.

D. SCIENTIFIC PROGRAMME

D.1 Scientific focus

The Action will advance the state of the art in runtime verification and its applications to important domains, making them applicable to a wider range of tasks in the design and implementation of computer systems. The focus is on the following directions:

Revisiting RV core questions

Fundamental theory and algorithms surrounding runtime verification will be revisited in the context of advances in other related fields such as static analysis, source code instrumentation, software testing, and formal verification. Furthermore, the current frontiers of runtime verification such as the issues involving diagnosing and mitigating failures, replaying of failure-leading traces, observations over multiple runs of a system, etc, will be reviewed and expanded. A taxonomy of dimensions of RV problems and solutions will be developed.

Consolidation of RV tools

The numerous runtime verification tools which are currently available will be reviewed, and classified, exploring ways in which related tools can interoperate and/or share mutual case studies, benchmarks, etc. Furthermore, the Action shall compare classes of the tools through the use of benchmarks, highlighting their respective strengths and weaknesses.

Challenging computation domains

Runtime verification has been studied in the context of several domains including embedded systems, hardware, distributed systems, and unreliable/approximate domains. To date such studies are fragmented and scattered at best. Through the Action, the existing work will be reviewed and a roadmap for future work will be drawn up.

Applying RV

While runtime verification has been evolving over a significant number of years, the number of industrial case studies is still on the low side. By bringing together the players in the field, the Action will facilitate the access and sharing of existing and novel case studies in areas including medical devices, financial systems, and legal contracts.

D.2 Scientific work plan methods and means

The Action will achieve its objectives through networking among runtime verification experts, and between runtime verification researchers and experts from application domains. The activities among experts will include the development of a common infrastructure that allows to compare, reuse and compose different tools.

The work of this Action will be structured according to the following four Working Groups.

Working Group 1: Core runtime verification

This group aims at clarifying the dimensions of runtime verification, its theory, algorithms and methods. These are the activities in which most of the work on RV has focused in the early stages of the discipline, with scattered results based on methods from other areas, notably formal methods and programming languages, and guided by application goals.

Many outcomes from the other working groups will pose new sets of problems and challenges for the core RV community.

Specific activities of Working Group 1 will include research actions centered around establishing a common framework for runtime verification, and challenges for new research and technology based on the other working groups. These activities include:

- To establish a taxonomy of runtime verification aspects that "paves the road" to allow a classification and comparison of theoretical results, problems and techniques.
- To identify the challenges and opportunities of instrumentation, where the system under scrutiny is modified or harnessed to allow the monitoring process.
- To study the interplay between runtime verification and static analysis, between runtime verification and model checking, and between runtime verification and testing. All these activities are proposed to increase or assess system's reliability, but their interplay can potentially increase their applicability. For example, by static analysis one can

specialize a system for a monitoring task and a monitor for a specific system, rendering a monitoring task feasible.

- To study applications of runtime verification beyond system observation. This includes reflection to act upon the system, typically to control and prevent errors, or to replay allowing an error to be reproduced or even fixed.
- To pose the challenges in monitoring quantitative and statistical data, beyond property violation.

The concrete output of this Working Group will be collaborative research papers, and a document describing a set of challenges and roadmap for each of the directions described above.

Working Group 2 : Standardization, benchmarks, tool interoperability

Working Group 2 aims to clear the landscape of formalisms and tools proposed and built for runtime verification, to design common input formats and to establish a large class of benchmarks and challenges.

Among the activities of this working group are the following:

- Enriching of the taxonomy of runtime verification concerns identified in Working Group 1 to classify tools and problems.
- The design of common representation formats for inputs to monitoring tools. The common language will not be a one-size-fits-all approach, as, for example, the definition of event and its data contents, and the dependence between a trace and the system that generates the trace varies widely. Instead, the outcome will be more a small family of suitable languages, following the taxonomy developed by Working Group 1
- The implementation of the interfaces in the common format for different programming languages and formalisms to enable that different RV tools can be attached to a variety of different programs and systems.
- The creation and maintenance of a collection of examples in the form of benchmarks, classified according to the taxonomy and expressed in the languages in the common format.
- The coordination of periodical activities for comparing tools against some of the benchmarks. This can take the form of a tool competition.

The concrete outputs of this Working Group will be a common computation infrastructure, including a description of the formats and implementations for programming languages of interest, and a collection of benchmarks.

Working Group 3 : Challenging computational domains

This group will study novel and challenging computational domains for runtime verification and monitoring that result from the study of other application areas than programming languages. The objectives of this Working Group will be to identify the challenges for monitoring in the following application domains:

- Distributed systems, where the timing of observations may vary widely in a non-synchronized manner.
- Embedded systems, where the resources of the monitor are constrained.
- Hardware, where the timing must be precise and the monitor must operate non disruptively.
- Unreliable domains and approximated domains, where either the system is not reliable, or aggregation or sampling is necessary due to large amounts of data.

These areas involve expertise from more than one domain and have a much higher chance of success if attacked cooperatively.

The concrete outputs of this Working Group will be twofold. First, a series of documents will be worked out giving a roadmap for the application of runtime verification techniques to the areas listed above, identifying connections with established work in the respective sub-areas of computer science, and challenges and opportunities. Second, a concrete case study will performed, in which a runtime verification solution for multicore systems will be developed using dedicated monitoring hardware based on FPGAs to show the feasibility and general applicability of runtime verification techniques.

Working Group 4 : Application areas (outside "pure" software reliability).

This group studies the potential applications of RV to important application areas beyond software and hardware reliability, including medical devices and legal contracts. This task requires the direct interaction with experts from the respective communities. The Working Group will organize workshops with invited experts from application domains.

For example, for the safe interoperability of medical devices, it is important to enrich the interface

specifications with temporal properties about the intended interaction of two devices and to synthesize monitoring code for runtime. If monitoring identifies unwanted behavior, the systems might go into some fail-safe mode.
Another interesting application area that will be explored is how to monitor legal e-contracts (e.g., computer-mediated transactions). Some efforts have recently been done to formalize legal contracts using formal languages, where skeletons of runtime monitors could be extracted from the formal semantics. The time is now right to apply these techniques successfully to real case studies, which is planned for this working group.
Other applications include robotic and hybrid systems, monitoring for business models and systems security.
Concrete output of this Working Group will consist on documents describing challenges and potential applications of runtime verification to these application areas. Moreover, a concrete case study in the medical domain will be performed identifying the safety enhancements of medical devices by using runtime verification techniques.

E. ORGANISATION

E.1 Coordination and organisation

The Action's organization is based on previous experience in successful actions, following the standard form of Rules of Procedure for Management Committee. The Action is coordinated by the Management Committee (MC), presided by Action Chair. Scientific activities are carried out through the Working Groups, led by Working Group Coordinators.
To promote the participation of young researchers, the Action places maximal emphasis in terms of its resources on Short-Term Scientific Missions (STSMs), devoted to PhD students and post-docs. The MC will appoint an STSMs committee formed by three MC members, and the specific guidelines for approval of STSMs. STSM grantees will be encouraged to present their progress in a subsequent WG meeting and workshop.
In the years in which the Action organizes a Summer School, the Action will support students from participating groups and countries as well as expert trainers, deciding in each case on an individual basis, based on the impact to accomplish the objectives of the Action.
The duration of the Action is four years. Action meetings include

- an organizational meeting of the MC and
- a scientific meeting with technical presentations of all Working Groups.

Technical presentations include results from the coordinated research and the insights from STSMs. Action meetings are organized in changing host countries. To foster the impact of the Action on the broader scientific community, Action meetings may be collocated with major conferences in the field.

In addition to the yearly meetings and STSMs, technical communication also proceeds through a new open online infrastructure. The MC will appoint at least one Action member to ensure maintenance of the Action web site and online system.

Each Working Group defines its specific milestones and summarizes progress towards the milestones in yearly reports and MC meetings.

Milestones.

The milestones of the Action correspond to the major decision points, and the moments at which the main objectives will be accomplished. These objectives include the reports produced by the WGs and the MC, the published material from the conferences and the training schools, and the tutorial materials. Specifically:

- M1 (end of Year 1): First Annual Report of the Action. Proposals on taxonomy, infrastructure, challenges. Layout of the industrial workshops for the rest of the Action. Report from application domain workshop.
- M2 (end of Year 2): Second Annual Report of the Action. Finalized taxonomy. First version of the electronic infrastructure and benchmarks. First document with challenges. Report from application domain workshop.
- M3 (end of Year 3): Third Annual Report of the Action. First Summer school. Second version of the infrastructure and benchmark, infrastructure for competitions. Report from application domain workshop.
- M4 (end of Year 4): Final report of the Action. Published surveys. Second Summer School. Final electronic infrastructure and competition. Report from application domain workshop. Report on case studies using monitoring hardware and application domains, including medical devices.

E.2 Working Groups

The Action consists of four Working Groups, with research plans outlined in Section D.2. The MC

will appoint two coordinators for each WG: a chair and a vice-chair.
The participants in the Action will become members of each working group depending on their interests. It is expected that many Action participants will belong to more than one WG, which will ensure the necessary collaboration between the groups.
The responsibilities of the coordinators of each Working Group include:

- the organization of the technical program at each WG meeting
- monitoring the progress of the research plan
- final preparation of relevant sections for yearly and final Action reports.

In particular, the coordinators of WG4 are in charge of organizing each of the workshops devoted to each of the application domains.

E.3 Liaison and interaction with other research programmes

Action experts participate in a number of activities sponsored by national and EU projects, as listed in Section B.4. When relevant to the Action objectives, experts and researchers from these projects that are not already active members of the Action will be invited to participate in Work Groups meetings. Along these lines, it is planned to organize one meeting jointly with researchers from the COST Action Betty (see B.4.) to exchange ideas.
One of the objectives of this Action is the interaction between runtime verification researchers and with domain experts, and these experts will be invited to oriented meetings to present their results and technical challenges.

E.4 Gender balance and involvement of early-stage researchers

This COST Action will respect an appropriate gender balance in all its activities and the Management Committee will place this as a standard item on all its MC agendas. The Action will also be committed to considerably involve early-stage researchers. This item will also be placed as a standard item on all MC agendas.
The Action complies with the European policy of equal opportunity between women and men as it is emphasized in the Treaty on European Union.
As described in Section E.1, the Action is specifically committing its resources to Short-Term Scientific Missions for PhD students. The Action will work to further encourage the participation of Early Stage Researchers and women by involving female PhD students in the Action projects.

Although there is a traditional imbalance towards the male sex in the computer science domain, the Action offers the possibility to incorporate application domains which attract more women. Among the experts interested in the Action are Early Stage Researchers and females. Early Stage Researchers and females were also already involved in preparing the COST Action proposal.

F. TIMETABLE

The duration of the Action will be 4 years. The Action will begin with a kick-off meeting during which the MC will be elected. The MC will then elect WG chairs, an STSM committee and a dissemination manager. There will be two WG meetings per year, coinciding with an MC meeting. There will a specialized workshop, each devoted to an application domain per year, collocated with one of the WG meetings.
We propose to organize Summer Schools in years 3 and 4.
The timetable is summarized in the following table:

	WG	MC	Summer School	Workshop
Year 1	2	2		1
Year 2	2	2		1
Year 3	2	2	1	1
Year 4	2	2	1	1

G. ECONOMIC DIMENSION

The following COST countries have actively participated in the preparation of the Action or otherwise indicated their interest: AT, CH, DE, DK, EE, ES, FR, MT, NL, NO, SE, TR, UK. On the basis of national estimates, the economic dimension of the activities to be carried out under the Action has been estimated at 52 Million € for the total duration of the Action. This estimate is valid under the assumption that all the countries mentioned above but no other countries will participate in the Action. Any departure from this will change the total cost accordingly.

H. DISSEMINATION PLAN

H.1 Who?

Target audiences for the dissemination of Action results are:

- the scientific community including especially the young investigators
- software and hardware engineers in industry, especially in medicine and finances
- lecturers in computer science
- the general public.

H.2 What?

The Action disseminates its results through

- leading competitive scientific publication venues
- the Action web site
- workshops on specific topics
- summer schools
- public lectures by Action members.

H.3 How?

Scientific results of the Action will be disseminated through refereed scientific journals and conference proceedings.
The Action web site itself will contain information on the scientific activities of the Action, including pointers to relevant information, standardized formats and descriptions of milestones reached and information for general public.
The involvement of domain experts from application domains will be done by devoting workshops to specific topics where both application experts and RV solution providers can discuss their work and cooperate on potential solutions.
Action members will incorporate the developed material into the courses they teach, facilitating the education of young investigators and helping the adoption of these techniques by the next generation of scientists and engineers. Summer schools will further serve to integrate students in the activities.