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

Secretariat

Brussels, 11 June 2009

COST 235/09

MEMORANDUM OF UNDERSTANDING

Subject :	Memorandum of Understanding for the implementation of a European Concerted Research Action designated as COST Action IC0901: Rich-Model Toolkit - An Infrastructure for Reliable Computer Systems

Delegations will find attached the Memorandum of Understanding for COST Action IC0901 as approved by the COST Committee of Senior Officials (CSO) at its 174th meeting on 26-27 May 2009.

MEMORANDUM OF UNDERSTANDING

For the implementation of a European Concerted Research Action designated as

COST Action IC0901

RICH-MODEL TOOLKIT - AN INFRASTRUCTURE FOR RELIABLE COMPUTER SYSTEMS

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 270/07 “Rules and Procedures for Implementing COST Actions”, or in any new document amending or replacing it, the contents of which the Parties are fully aware of.

2. The main objective is making automated reasoning techniques and tools applicable to a wider range of problems, as well as making them easier to use by researchers, software developers, hardware designers, and information system users and developers.

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 44 million in 2009 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 Chapter V of the document referred to in Point 1 above.

TECHNICAL ANNEX

A. ABSTRACT AND KEYWORDS

The Action will coordinate the development of infrastructures for automated reasoning about Rich Models of computer systems. Rich Models have the expressive power of all practically formalizable mathematics, enabling natural specification of software, hardware, embedded, and distributed systems. Rich Models support modeling at a wide range of abstraction levels, from knowledge bases and system architecture, to software source code and detailed hardware design.

The Action contributes to the construction of a Rich Model Toolkit, which is a unified infrastructure that precisely defines the meaning of Rich Models, standardizes representation formats, and supports a number of automated reasoning tools. Moreover, the Action develops and deploys new tools for automated reasoning that communicate using these standardized formats. The goal of resulting tools is to have a wide range of applicability and improved efficiency, helping system developers construct reliable systems through automated reasoning, analysis, and synthesis.

Keywords: Automated reasoning, Verification, Synthesis

B. BACKGROUND

B.1 General Background

Researchers have recently developed a number of useful tools for automated analysis of particular classes of models of computer systems:

- Hardware manufacturers are using SAT solvers, model checkers, and theorem provers to identify and correct errors that could have enormous financial consequences.
- Software vendors are using static analyses supported by automated theorem provers and constraint solvers to prevent software crashes.
- Description logic reasoners analyze relationships between tens of thousands of terms in medical ontologies and verify their consistency.
- Aircraft manufacturers and space agencies are using analysis tools based on abstract interpretation to eliminate errors in aircraft control software.

Despite these individual successes, today’s automated analysis methods are not widespread in engineering practice, and therefore have a limited impact. Among the factors contributing to this state of affairs are the limitations of the tools themselves: the lack of automation, specialized input formats, and a limited use of high-level synthesis, which makes these tools expensive to use in practice. Another factor is the social circumstances, such as the lack of quality standards that differentiate formally certified computer systems from systems without formal assurance guarantees.

To address these problems, the Action makes a dedicated effort to unify current specialized algorithms and hide their internal complexity from users. Among the central ideas in these activities is the notion of Rich Models, general and rigorous representation of different types of computer systems, different system aspects, and different levels of detail (from design to implementation, from functional correctness to timing and performance). Driven by this notion, the Action coordinates the development of automated analysis and synthesis tools, ensuring that the tools accept models expressed in the same general-purpose language, and enabling communication between the tools.

A number of national programs support research activities of the Action experts. Without the Action, however, these research activities would be carried out independently, would involve duplication of effort, and would not benefit from timely exchange of ideas. The strong need for coordination, as well as the fundamental and broad nature of the research involved means that no research program other than COST is appropriate to support such unified effort at the current, initial, coordination stage. The COST framework is ideal for the proposed coordination. On the one hand, the development of representation formats for Rich Models requires discussions among all the parties involved, which is best done through joint COST meetings. On the other hand, the core technical work of the Action occurs in individual research groups, and Short-Term Scientific Missions for Early Stage Researchers are ideal for the underlying intensive technical interaction.

B.2 Current state of knowledge

This section reviews current state of knowledge in automated reasoning according to the type of reasoning technique. It concludes with an overview of related standardization efforts that have made such techniques more widely applicable.

SAT solvers, such as zChaff, Berkmin, and Minisat, solve the satisfiability problem for propositional logic. The progress in SAT technology in the past decade demonstrated that worst-case computational complexity need not prevent practical applications of tools in verification of hardware and software.

Finite model finders, such as Paradox, Kodkod, and Darwin search for finite structures that satisfy given first-order logic formulas. They often use a reduction to SAT. They have proved very useful in finding counterexamples to conjectures, including those that involve properties of software implementations and designs.

First-order theorem provers find proofs of validity of given first-order logic formulas (in contrast to model finders that find counterexamples). Decades of research into resolution techniques resulted in implementations such as SPASS, E and Vampire, which are very effective at proving a wide range of valid first-order logic formulas. Promising areas of research are methods that combine proof search and model finding, improving the effectiveness of both approaches.

Description logics avoid the undecidability of first-order logic by adopting a form of bounded quantifiers. Reasoning tools based on decidable description logics, such as FaCT++, Racer, and Hermit, have proved capable of handling complex formulas, while scaling to problem instances with a large number of background axioms. This progress was enabled by systematic study of decidability, complexity, and practical reasoning algorithms for specific classes of formulas.

Decision procedures of practical importance include the MONA tool for Weak Monadic Second-Order logic over trees, applied e.g. to verification of linked data structures. Widely applicable are also arithmetic decision procedures, which incorporate insights from linear programming. Recently, decision procedures for reasoning about sets and multisets have been built on top of arithmetic decision procedures. Overall, specialized reasoning techniques have seen both theoretical advances and implementations in the context of more general reasoning systems.

Satisfiability modulo theories (SMT) is among such general reasoning techniques that show great promise in combining multiple specialized decision procedures into a decision procedure for a richer language. SMT solvers build on techniques such as Nelson-Oppen combination of disjoint theories. Recent advances include SAT-based frameworks such as DPLL(T), and successful standardization activities such as the SMT-LIB initiative.

Hardware and software verification research has benefited from the advances in SAT and SMT reasoning, and also developed specific techniques for reasoning about transition systems. As in reasoning about first-order formulas, for large state spaces we can distinguish model finding methods (bounded-exhaustive testing, explicit state model checking), and proof-based methods (data-flow analysis). The success of automated reasoning about safety and liveness of hardware systems is witnessed by routine use of these techniques within major microprocessor manufacturers, and was acknowledged by the recent Turing award. Recently, techniques based on predicate abstraction, bounded model checking, transition invariants, and symbolic shape analysis brought together research in transition systems and research in decision procedures for formulas, resulting in tools for automated analysis of transition systems with large state spaces.

Abstract interpretation is an influential technique for analyzing transition systems with large state space. Abstract interpretation tools such as ASTREE and AbsInt have found use in the analysis of avionics software. Such analyzers are based on parameterized abstractions and have recently made use of advances in decidable classes of logical constraints, such as difference logics. Among the goals of the Action is taking further such fruitful combinations and identifying similar cross-fertilization opportunities between other sub-fields of automated reasoning.

Automated synthesis of executable systems from specifications is among the most ambitious approaches to reliable computer systems. Most of the above automated reasoning tools can provide not only simple yes/no answers but also concrete counterexamples, proofs, and reachability invariants. An active area of research is productively exploiting such information for constructing reliable computer systems. Recent exciting practical applications include specialized software synthesis approaches and hardware synthesis from linear temporal logic specifications.

Standardization activities, benchmarks, and competitions have tremendously accelerated the development of automated reasoning tools.

- DIMACS format for SAT, with the SAT Competition and the SAT Race, are often credited with sparking great advances in SAT solving.
- The SMT-LIB initiative defined standard input/output formats and interfaces to SMT solvers. Over a period of several years it became supported by a number of SMT solvers, and a number of formal methods client applications. The initiative runs an influential yearly competition, SMT-COMP, and contains a growing repository of over 40,000 benchmarks from academia and industry.
- The TPTP format for first-order logic has long provided stable interfaces to powerful first-order provers and model finders. It has recently been extended to higher-order logic, and it continues to evolve.

Such standardized formats will become even more successful if they support natural descriptions of a wider range of problems. The starting point for the Rich Model Language designed as part of the Action are therefore the expressive languages of interactive provers such as Isabelle, HOL, and Coq. On the *syntactic* side, related initiatives include OMDoc, whose goal is the representation of mathematics on the Web, and recently the higher-order TPTP format. The key property of concrete interactive theorem prover languages is their clear and well-understood *semantics*, as well as the number of defined library concepts from computer science and mathematics. The adequacy of these library concept definitions is empirically proven through formalizations such as the correctness proofs for Java infrastructure, correctness of SAT solvers, meta-theorems of first-order logic, deep results in set theory, the proof of four-color theorem in graph theory, and proofs of important steps of the Kepler conjecture. The process of formalization continues through new submissions to the Archive of Formal Proofs (http://afp.sourceforce.net), the Verisoft project, and the POPLmark challenge. In the area of the transition system analysis, the conference on Computer-Aided Verification recently introduced a competition for hardware verification with a standardized input format. In the area of software, intermediate formats with precise semantics include SAL from SRI and BoogiePL from Microsoft Research. A striking observation about current standardized formats is that they either have limited expressive power or limited support for automated reasoning. To the extent possible, this Action aims to obtain both expressive power and automation, by embedding existing languages into a unified Rich Model Language, and by developing new algorithms and tools that fill the automation gaps between current specialized approaches.

Innovative activities of the Action include all areas summarized in this section. They involve theoretical work on decidability, complexity, and algorithm design, as well as tool development and computing experiments on models of practical interest. Overall, they contribute to wider applicability of automated reasoning about computer systems.

B.3 Reasons for the Action

Reasons for the Action are coordinating automated reasoning research to make the techniques and tools more powerful and easier to use by developers of computer systems, including hardware, software, large-scale information systems, and data centers.

The notion of Rich Model Language will be more expressive than any of the existing languages used to describe analyzable models. This expressive power will enable a broad community to agree to use the Rich Model Language. Researchers and developers will be able to directly represent the implementations of software and hardware systems, avoiding manual abstraction and lowering the expertise needed to construct sophisticated analysis and synthesis tools. The Rich Model Language will also foster research in analysis and synthesis algorithms because it will enable researchers to compare a wide range of techniques on a collection of models of practical interest, leading to the exchange of fruitful ideas across different approaches.

B.4 Complementarity with other research programmes

The research topics of the Action are currently supported by over 15 independent national programs. Further ongoing efforts that are synergistic with the planned activities include: HATS (Highly Adaptable and Trustworthy Software using Formal Models), MOBIUS, High Integrity Java, GAMES ESF Research Networking Programme, COST Action IC0701 on Formal Verification of Object-Oriented Software, TYPES FP6 Project no. 51099, FP6 STReP Prosyd project, FP7 STReP COCONUT project, AVANTSSAR FP7 project, ARTIST2/ARTIST Design Network of Excellence.

A relevant world-wide initiative compatible with a fraction of the Action goals is the Verifying Compiler Grand Challenge for Computing Research. Even more relevant are past EU projects on integration of reasoning techniques, including the PROSPER toolkit, funded under the ESPRIT program. Related activities in the United States include the integration of formal method tools in the SRI Computer Science Laboratory (for example, the Evidential Tool Bus proposal), and the Bandera tool set at the Kansas State University.

The activities listed above are a source of particular classes of models and specialized algorithms. This Action will include and collaborate with researchers involved in these activities. However, no prior activity by itself proposed such a general notion of Rich Models, and aimed at unifying such a broad set of automated reasoning techniques.

C. OBJECTIVES AND BENEFITS

C.1 Main/primary objectives

The main objective of the Action is making automated reasoning techniques and tools applicable to a wider range of problems, as well as making them easier to use by researchers, software developers, hardware designers, and information system users and developers.

C.2 Secondary objectives

Secondary objectives are:

- increasing the coherence, visibility, and competitiveness of automated reasoning research
- assessing the potential for industry standards that certify the added value of computer systems developed using automated reasoning technology

C.3 How will the objectives be achieved?

The objectives will be achieved through:

- Working Group meetings
- Short-Term Scientific Missions
- information exchange with industry
- initiation and maintenance of an online forum and an online reference for the area of automated reasoning
- publications of results in leading computer science conferences and journals
- organization of a tool competition in the area of automated analysis, synthesis, and certification of Rich Models
- training of PhD students through advanced seminars.

C.4 Benefits of the Action

In addition to the inherent benefits from coordination of research and cross-fertilization of ideas in different domains, concretes outcomes of the Action will include the Rich Model Toolkit, a set of infrastructures connected through the Rich Model Language, including a set of communicating automated reasoning tools. The tools will help developers construct reliable systems by automatically analyzing and synthesizing systems and their components.

The automation level of tools in the Rich Model Toolkit will make tool adoption COST-effective, resulting in higher-quality computer systems, and improving the safety and availability of information technology used by all members of the society. These efforts will also reduce the likelihood of future disasters such as the airline and aerospace failures, and avoid further microprocessor and automobile recalls due to software bugs.

C.5 Target groups/end users

In addition to the research community, target groups and end users include: developers and designers of software, hardware, and embedded systems, educators, industrial organizations, and students.

Developers and designers of computer systems will directly benefit from the sophisticated tools in the Rich Model Toolkit developed in the course of the Action. These tools will detect errors in designs and implementations, repair errors, and synthesize new implementations from specifications. The developers using such tools will be more productive and will be able to focus more on the creative and domain-specific aspects of their work.

Educators will be able to motivate and illustrate the theory of reasoning about computer systems through working tools usable by students, and concrete examples from the practice.

Industry will be able to use the Rich Model Toolkit in system development, increasing the competitiveness and reducing the cost. Moreover, unified formats and new algorithms resulting from the Action will provide guidance for technology transfer, enabling the development of a new generation of industrial tool products based on the principles of the Rich Model Toolkit.

D. SCIENTIFIC PROGRAMME

D.1 Scientific focus

The Action will advance algorithms and implementations of automated reasoning technology, making them applicable to a wider range of tasks in the design and implementation of computer systems. The focus is on the following directions:

- Design of the Rich Model Language, taking into account the ease of modeling, current specialized languages, requirements of existing and new algorithms and tools, and interactive theorem proving language semantics.
- The adaptation of existing tools and the development of new tools that support the Rich Model Language, including both tools for automated reasoning about Rich Models (deduction, analysis and synthesis), and tools for automatically generating Rich Models.

- Development of new algorithms applicable across different application areas, as long as the input Rich Model has the mathematical structure supported by the algorithm.
- Techniques that compose multiple specialized algorithms into algorithms applicable to a wider range of problems, with understood guarantees on the soundness, completeness, and efficiency.
- Computer experiments with automated reasoning about Rich Models from the area of software, hardware, embedded, and information system verification.

The pace of the activities will crucially depend on the ability to develop and share the expertise among Action participants. The research (and dissemination) activities will be carried out by researchers using state-of-the art commodity computing equipment (desktops, compute servers, web servers).

The research will be carried out in at least the following cross-cutting and collaborating Working Groups:

1. Rich Model Language: design and benchmark Suite
2. Decision procedures for Rich Model Language fragments
3. Analysis of executable Rich Models
4. Synthesis from Rich Model Language descriptions.

D.2 Scientific work plan: methods and means

The Action will achieve its objectives through the development of foundations and tools of the Rich Model Toolkit infrastructure. The infrastructure will consist of a collection of Rich Models, and a system of tools communicating using the Rich Model Language. The work will be structured according to the following four initially-envisioned Working Groups.

Working Group 1: Rich Model Language Design and Benchmark Suite

Rich Model Language and benchmarks written in this language are among concrete results and unifying themes of the Action. They are also the focus of Working Group 1. The specific activities of Working Group 1 will include the following.

- *Design of the Rich Model Language*, including abstract and concrete syntax, as well as semantics; inspired by the expressiveness of provers such as Isabelle/HOL while aiming for simplicity of automated processing present in more specialized languages.
- *Translations* between Rich Model Language and languages such as Isabelle/HOL, SMT-LIB, TPTP, and OWL.
- Design of *formats* for expressing manually and automatically constructed *proofs and counterexamples* for properties of Rich Models, as well as implementing efficient and trustworthy checkers and visualizers for these formats.
- *Building of Rich Model benchmark collection* for comparing different tools and measuring progress in tool development, made publicly available on the Web.
- Helping *adapt existing tools* to take advantage of Rich Model infrastructure.
- Realistic plans for running Rich Model *tool competition***.**

The tools in the Rich Model Toolkit will accept a set of Rich Models and produce a new set of Rich Models (with the output in a formally verifiable relationship to the input). This general view supports not only the traditional validity and satisfiability checking, but also optimization and synthesis problems.

The Action will build on the experience from the following past successes of its experts:

- initiating *successful community standards* for automated reasoning tools
- leading major efforts in *proof-assistant development*
- developing *specialized automated reasoning tools*
- *automated generation of models* from applications such as hardware and software verification.

The Action is therefore in a unique position to develop Rich Model Language format and to advertise it within the scientific community, which in turn will foster the adoption of the format in industry.

Working Group 2: Decision Procedures for Rich Model Language Fragments

Working Group 2 focuses on automating the reasoning about Rich Models through development, analysis, implementation, formal verification, and applications of decision procedures. A decision procedure accepts a class of rich models representing logical formulas and (within well-understood time and space bounds) provides an answer about the validity of the formula. Decision procedures of interest in the Working Group include decision procedures for sets, collections with cardinality bounds, relations, arrays, bit vectors, transitive closure logics, non-linear arithmetic, and description logics. Among the topics of interest are the following.

- *The improvement of efficiency of existing decision procedures.*
- *The development of new decision procedures.*
- *Integration* of decision procedures into satisfiability modulo theory *(SMT)* and *resolution* frameworks.
- Automated *synthesis of decision procedures*, in collaboration with Working Group 4.
- *Modular, flexible, and efficient implementations* of SAT and SMT solvers, including: proving validity of Rich Models using decision procedures, finding counterexamples, solving optimization problems, supporting the use of off-the-shelf SAT/SMT solvers through converters between the Rich Model Language and DIMACS and SMT-LIB input formats, extensible SMT solver architectures that support multiple background theories (such as Nelson-Oppen and DPLL(T) combination), and efficiency improvements in SMT and SAT (including non-clausal SAT and SMT solvers).
- *Applications of SAT and SMT solving* to real-world decision and optimization problems, including hardware verification, software verification, planning, scheduling, and timetabling.
- Improving techniques for *encoding real-world problems into SAT and SMT*, including 1) the design and implementation of high-level Rich Model Language support that enables natural problem description and leaves room for efficient choice of encoding, 2) choosing appropriate background axioms and controlling quantifier instantiation, and 3) automation of the encoding process starting from high-level Rich Models.

- *High-confidence implementations of decision procedures,* including: extending solvers to generate evidence (models for satisfiable formulas and proofs for unsatisfiable formulas), applying software verification techniques to verify the calculi and parts of implementations of SAT and SMT solvers, development of verified implementations of quantifier-elimination procedures, and exploring the role of synthesis in obtaining provably correct implementations.
- Scalable reasoning in expressive description logics with applications to medical ontologies (SNOMED, NPfIT, GALEN, NCI Thesaurus, OBO Foundry), software systems, Semantic Web, e-Science, and the Grid.

Through these activities, Working Group 2 will contribute to the development of efficient and reliable automated reasoners for a significant class of practically relevant Rich Models.

Working Group 3: Analysis of Executable Rich Models

Working Group 3 focuses on the analysis of dynamic state changes in systems such as software systems, hardware designs, embedded systems, and communication protocols. Such changes can be described by a general notion of a *transition system*. Transition systems are therefore an important class of Rich Models, with both exact semantics and a mapping to physical implementations.

To address the decidability and complexity limitations of the general problem, Working Group 3 focuses on

- *abstraction-based approaches* that provide semi-algorithms for the general analysis tasks
- efficient *algorithms for the specialized subclasses*.

Properties considered include both safety (reachability) and liveness (termination). The group aims to develop theory, algorithms and implementations for verification of transition systems by leveraging the expertise across the areas of abstract interpretation, automated deduction, and constraint solving. Specific sub-problems considered include the following.

- Developing *refinement techniques* and tools that deal with expressive data types such as lists, trees, and their combination with arithmetic.

- Developing abstraction-based analysis techniques and tools suitable for finding both *proofs and witnesses for property violation.*
- Combining precise (but potentially slow) *predicate abstraction* techniques with fast (but potentially imprecise) *specialized analyses* to reduce the number of abstraction refinement iterations and speed-up the analysis.
- Exploring synergy between *synthesis methods* in Working Group 4 and the *invariant/ranking function generation* techniques used for transition system analysis. This ambitious direction will exploit the duality of synthesis and analysis to deliver better theoretical insight and automatic tool support for both tasks.
- Integration of *data-flow analysis* algorithms, shape analyses, SAT, SMT and BDD-based *model checking***,** *symbolic execution* and *bounded model checking* into the Rich Model Toolkit.
- Integration of *state/event-based formalism* into Rich Models
- *Synergy with SMT solver technology* of Working Group 2 to improve overall search efficiency.
- Developing *tools that extract Rich Models* from software source code, software bytecode, and hardware designs, with applications to analysis of: functional programs, linked and concurrent data structure implementations, correct resource use and finite-state protocols.
- Automated *detection of security flaws***,** attacks, intrusions, and violations of user-specified security policies.

Through a combination of tools that *extract* Rich Models and tools that *analyze* Rich Models, Working Group 3 will enable automated analysis of expressive properties of computer systems, directly helping computer system developers.

Working Group 4: Synthesis from Rich Model Language Descriptions

Working Group 4 explores the theory, tools, and usability of *synthesis* in system development. In contrast to automated *verification* algorithms, which establish whether a given system satisfies a given specification, synthesis algorithms automatically construct implementation that is guaranteed to satisfy a given specification. Synthesis is more difficult than verification, and is one of the holy grails of Computer Science. Despite impressive theoretical results of the past, it was only recently that researchers made significant steps towards the development of practical synthesis algorithms. Synthesis still has many limitations preventing its wider practical application. Working Group 4 aims to address these limitations through tasks that include the following.

- Developing *synthesis algorithms for more expressive logics***,** including identifying decidability and worst-case complexity of synthesis for expressive logics, developing heuristics and new subclasses of problems that overcome high complexity, lifting decision problems (explored in Working Group 2) to synthesis problems, developing high-level synthesis techniques applicable to components, and synthesis of hybrid systems.
- *Efficient implementations of synthesis* algorithms using not only binary decision diagrams but also quantified (Boolean) formulas, and the development of benchmarks for synthesis problems within the Rich Model benchmark suite from Working Group 1.
- *Quantitative generalization of synthesis***,** including: extending Boolean specifications by quantitative measures in order to rank implementations by laziness, fairness, or parsimonious use of resources, non-Boolean algorithms that generalize decision procedures from Working Group 2, and quantitative games as a method for solving synthesis problems.
- *Simplified synthesis problems of practical interest***,** including problems with limited quantifier alternations, using synthesis for problems where some part of the structure is predefined (e.g., repair, sketching).
- Using synthesis to implement *high-level programming language constructs***.**

Action experts have a unique set of complementary skills, whose combination will be necessary to fulfill the research vision of synthesis. By giving a more active role to automated tasks and avoiding low-level coding, synthesis has the potential to dramatically improve the productivity of computer system developers.

E. ORGANISATION

E.1 Coordination and organisation

Organization of the Action follows 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) for PhD students. The MC appoints a STSMs coordinator and the specific guidelines for approval of STSMs by the MC. To maximize the resources available for STSMs, the Action has one meeting per year. Continuous communication occurs through STSMs and an organized online forum.

The duration of the Action is four years. Yearly Action meetings include

- an organizational meeting of the MC and
- a scientific meeting (Rich Model Conference) with technical presentations of all Working Groups.

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

In addition to the yearly meetings and Short-Term Scientific Missions, technical communication also proceeds through a new open online forum. The MC appoints at least one Action member to ensure maintenance of the Action web site, and at least one member to ensure the maintenance and the persistence of the online forum.

Each Working Group defines its specific milestones and summarizes progress towards the milestones in yearly reports and MC meetings. A central activity of the Action is the design of the Rich Model Language format. A first draft of the format and a basic set of support tools are expected by the end of year two of the Action. The integration of a number of specialized reasoning tools is expected by the end of the Action.

E.2 Working Groups

The Action contains four Working Groups, with research plans outlined in Section D.2. In the course of the Action up to two additional Working Groups will be introduced if needed, according to the interest of current and newly included Action members.

The responsibilities of each Working Group Coordinator include

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

E.3 Liaison and interaction with other research programmes

Action experts participate in a number of activities sponsored by national and EU projects. Through its experts, the Action actively interacts with the related domain-specific research. The experts regularly present research findings from the application areas at the Working Group meetings, and communicate the relevant benchmarks.

As a specific example of the nature of this interaction, we point out the simultaneous synergy and non-overlap with the area of software verification. A number of software verification tools focus on expressive programming language constructs and programming methodology, such as object-oriented methodology, but often use automated reasoning techniques as a black box, without developing automated reasoning techniques themselves. The Action will develop such automated reasoning techniques, taking into account the needs of several application domains. The results of the Action will therefore eliminate the bottlenecks currently experienced by software verification tools, especially in the area of verifying strong properties that ensure correct functioning of software.

In addition to software verification, important application domains where the Action has significant expertise include hardware verification and synthesis, formalized mathematics, reasoning about medical terminologies, and the Semantic Web. Each of these areas has its own community of researchers, and a strong economic dimension of its own. What is common to all of them is the need for automated reasoning expertise. The unification of such expertise across different application domains is one of the justifications for the present Action.

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.

Among the experts interested in the Action, at least 7 are Early Stage Researchers and at least 4 are female. Women also play key roles in the organizations participating in the Action. The Action will work to further encourage the participation of Early Stage Researchers and women by involving female PhD students in the Action projects.

F. TIMETABLE

The duration of the Action is 4 years and begins with the kick-off meeting. At the kick-off meeting the MC will appoint Action Chair, as well as coordinators for STSMs, Web site, online forum, and the Working Groups. MC will also take immediate steps towards quick approval of the initial set of STSMs.

The Action will have exactly one meeting per year, in order to maximize the resources dedicated to STSMs. Assuming the set of experts that have so far expressed interest in the Action, 20 STSMs are expected in each of the years. This number will need to be increased if the actual membership is larger.

G. ECONOMIC DIMENSION

The following COST countries have actively participated in the preparation of the Action or otherwise indicated their interest: AT, CH, CZ, DE, ES, FR, IL, IT, RO, RS, UK. On the basis of national estimates, the economic dimension of the activities to be carried out under the Action has been estimated at 44 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
- standardization bodies
- teachers and educators in computer science
- the general public.

H.2 What?

The Action disseminates

- the Rich Model Language definition
- insights from integrating different automated reasoning approaches
- insights into new specialized techniques
- tool descriptions
- entries in a new online reference
- essential technical results in the field of automated reasoning
- high-level overviews of the impact of research on the general public.

H.3 How?

The Action disseminates its results through

- leading competitive scientific publication venues
- technical reports
- rapid communications on the online forum introduced by the Action
- the Action web site
- courses taught by Action members
- public lectures by Action members
- yearly Action meetings, with selected and representatives of industry and educational institutions.

On the online forum, Action members will continuously and efficiently present technical insights of the community. The forum will support stable citation and work attribution. It will be open to the public, but linked to the Action web site. The initial editorial board of the forum is selected from Action members.

A reference collection of articles describing main theoretical results in automated reasoning (a form of online encyclopedia) will be linked to the online forum. The Action initiates the encyclopedia during the Action by producing a critical mass of articles, with the hope of turning it into a community effort with high-quality knowledge from the field.

The Action web site itself will contain information on the scientific activities of the Action, including pointers to relevant information, standardized formats and benchmarks, descriptions of milestones reached, news from the automated reasoning community, and information for general public including press announcements.

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.