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

Secretariat

Brussels, 16 January 2009

Full proposal reference oc-2008-2-2705 for a COST new Action

Subject: Full proposal for a new COST Action:
Rich-Model Toolkit: An Infrastructure for Reliable Computer Systems

Proposer: Prof. Viktor KUNCAK
Swiss Federal Institute of Technology
EPFL / I&C / LARA / INR318
Station 14

CH
viktor.kuncak@epfl.ch

National Coordinator: [*]

Domain Committee: Information and Communication Technologies

[*] Will be completed by the COST Office

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

COST Action

Rich-Model Toolkit: An Infrastructure for Reliable Computer Systems

The signatories 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 299/06 "Rules and Procedures for Implementing COST Actions", or in any new document amending or replacing it, the contents of which the Signatories are fully aware of.

2. The main objective of the Action is [*]

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 [*] Euro [*] million in [*] prices.

4. The Memorandum of Understanding will take effect on being signed by at least five Signatories.

5. The Memorandum of Understanding will remain in force for a period of 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.

[*] Will be completed by the COST Office

A. ABSTRACT & KEYWORDS

A.1 ABSTRACT

The Action coordinates activities on developing infrastructures for automated reasoning about the new notion of Rich Models of computer systems. Rich Models have the expressive power of a large fragment of formalizable mathematics, enabling 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 Rich-Model Toolkit, an new unified infrastructure that precisely defines the meaning of Rich Models, introduces standardized representation formats, and incorporates a number of automated reasoning tools. Moreover, the Action develops and deploys new tools for automated reasoning that communicate using these standardized formats. The resulting tools will have a wide range of applicability and improved efficiency, helping system developers construct reliable systems through automated reasoning, analysis, and synthesis.

A.2 Keywords

automated reasoning, automated theorem prover, decision procedure, 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 subtle errors that could have enormous financial consequences.
- Software vendors are using static analyses supported by state-of-the art automated theorem provers and constraint solvers to prevent software crashes.
- Static analysis tools can analyze software source code, automatically constructing mathematical models of millions of pages length, proving non-interference of system components and detecting safety and security errors.
- 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 successes, today’s automated analysis methods are not widespread in engineering practice, and therefore have a limited impact. One factor 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 are the social circumstances, such as the lack of quality standards that would differentiate formally certified computer systems from systems without formal assurance guarantees.

To address these problems, the Action makes a conscious effort to unify current specialized algorithms and hide their internal complexity from the users. Among the central ideas in these activities is the notion of Rich Model, a precise and universal 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. Moreover, the Action coordinates research on concrete algorithms and tools for reasoning about the particular automated reasoning problems.

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 this stage. The format of COST activities is ideal for the proposed coordination. The development of representation formats for rich models requires discussions of the parties involved, which is best done through joint COST meetings. 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 necessary intense technical interaction.

B.2 Current state of knowledge

This section reviews current state of knowledge in automated reasoning according to the type of the 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. They made great progress in the past decade, showing that worst-case computational complexity does not prevent practical applications of tools in verification of hardware and software, and invigorated research in automated reasoning about more expressive classes of problems.

Finite model finders, such as Paradox and Kodkod, search for finite structures that satisfy given first-order logic formulas, often based on a reduction to SAT. They proved very useful in finding counterexample 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 variants of resolution resulted in implementations such as SPASS, E, and Vampire, that 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 overcome 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. Such progress was made possible by systematic study of decidability, complexity, and practical reasoning algorithms for specific classes of formulas.

Decision procedures of practical importance also include the MONA tool for weak monadic second-order logic over trees, which found important applications such as verification of linked data structures. Also relevant are arithmetic decision procedures and procedures for reasoning about sets and multisets, which have incorporated insights from linear programming. Such specialized reasoning techniques have recently seen both theoretical advances and implementations in the context of more general reasoning systems.

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

Hardware and software verification has benefited from the advances in SMT reasoning, and helped identify the area of reasoning about transition systems. Just as in reasoning about first-order formulas, we can distinguish model finding methods, such as bounded-exhaustive testing and explicit state model checking, and proof-based methods. The success of automated reasoning about safety and liveness of finite-state systems is witnessed in routine use of these techniques by major microprocessor manufacturers, acknowledged by the most 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 infinite-state transition systems.

Abstract interpretation is a related techniques for analyzing infinite-state systems. Abstract interpretation tools have recently seen great successes, such as the use of ASTREE and Absint tools 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 techniques 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.

- TPTP formats for first-order logic that have long provided stable interfaces to powerful provers.
- DIMACS format for SAT, with the SAT Competition and the SAT Race is 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.

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. The key property of concrete interactive theorem provers 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, metatheorems 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 formalizations continues through new submissions to the Archive of Formal Proofs (http://afp.sourceforce.net) and efforts such as 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 include 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 proposed 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 FP6/FP7 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 that include 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 include

- 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

- work 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
- organizations of 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 centered around the Rich-Model Language that includes 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
- 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 aspect of their work.

The 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 tools in the 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 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 of research includes the following directions as well as the topics that are naturally connected or build on them.

- Rich Model Language design that takes into account the ease of modelling, 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 Rich Model Language, including both tools for automated reasoning about Rich Models (deduction, analysis and synthesis), and tools for automatically generating Rich Models
- new specialized algorithms, not limited by any particular application area, as long as the underlying 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 Work 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 Work Groups.

D.2.1 Work 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, and are the subject of Work Group 1. The activities of Work Group 1 will include

- **Rich-Model Language design**, 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
- **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 Rich-Model Toolkit will accept a set of Rich Models and produce a new set of Rich Models (with 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 similar formats in industry.

D.2.2 Work Group 2: Decision Procedures for Rich Model Language Fragments

Work Group 2 focuses on automating the reasoning about Rich Models through development, analysis, implementation, formal verification, and applications of decision procedures. A decision procedure typically 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 Work 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 Work 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 sDIMACS and SMT-LIB input formats, extensible SMT solver architectures that support multiple background theories (such as Nelson-Oppen and DPLL(T1,...,Tn) combination), 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 the design and

implementation of high-level Rich Model Language support that enables natural problem description while leaving space for efficient choice of encoding, choosing appropriate background axioms and controlling quantifier instantiation, and 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 entire 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, Work Group 2 will contribute to the development of high-assurance efficient automated reasoners for a significant class of practically relevant Rich Models.

D.2.3 Work Group 3: Analysis of Executable Rich Models

Work Group 3 focuses on the analysis of dynamic 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, Work 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 subproblems 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 Work 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 Toolit
- integration of **state/event-based formalism** into Rich Models
- **synergy with SMT solver technology** of Work 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, Work Group 3 will enable automated analysis of expressive properties of computer systems that directly help developers construct reliable software.

D.2.4 Work Group 4: Synthesis from Rich Model Language Descriptions

Work Group 4 explores the theory, tools, and usability of *synthesis* in system development. In contrast to automated *verification* algorithms that establish whether a given system satisfies a given specification, synthesis algorithms automatically construct the 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. Work 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 rich logics, developing heuristics and new subclasses of problems that overcome high complexity, lifting decision problems (explored in Work 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 Work 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 Work 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 will follow the standard form of Rules of Procedure for Management Committee. The Action will be be coordinated by the management Committee (MC), presided by Action Chair. Scientific activities will be carried out through the Work Groups.

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 MC. To maximize the resources available for STSMs, the Action is expected to have exactly one meeting each year. Continuous communication will occur through STSMs and an organized online forum.

The Action will last for four years. Yearly Action meetings will include

- organizational meeting of the MC and
- technical presentations of all Work Groups

Technical presentations will include results from the coordinated research and the insights from short-term scientific missions. Yearly Action meeting will be organized in changing host countries. To foster the impact of the Action on the broader scientific community, each Action meeting will be collocated with a major conferences in the field.

In addition to the yearly meetings and short-term scientific missions, technical communication will also proceed through a new open online forum. The MC will appoint 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 Work Group will define its specific milestones during the first year and will summarize progress towards the milestones in yearly reports and MC meetings. A central activity of the Action is the Rich Model Language format. We expect a first draft of the format and a basic set of support tools to be available by the end of year two of the Action. We expect the integration of a number of specialized reasoning tools by the end of the Action.

E.2 Working Groups

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

The responsibilities of each Work 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 will participate in a number of activities sponsored by national and EU projects (see Section B.4). Through its experts, the Action will therefore actively interact with the related domain-specific research. The experts will regularly present research finding from the application areas at the Work 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 rich programming language constructs and programming methodology, such as object-oriented methodology, but use automated reasoning techniques as a black box, without developing automated reasoning techniques themselves. The proposed Action develops 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.

However, 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, and coordinators for STSMs, Web site, online forum, and Work 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 (twenty) 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,CZ,FR,DE,IL,IT,RO,RS,ES,CH,UK. On the basis of national estimates, the economic dimension of the activities to be carried out under the Action has been estimated at 11 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.

Assuming

- 11 participating countries
- 10 participants for each country
- 100'000 Euro per participant for the duration of the Action

the multiplication provides the above estimate of 11 Million Euro for the total duration of the Action.

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
- general public.

H.2 What?

The Action will disseminate

- 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 will disseminate its results through

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

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

Linked to the online forum will be a reference collection of articles describing main theoretical results in automated reasoning, a form of online encyclopedia. We expect to initiate the encyclopedia during the Action by producing a critical mass of articles, then turn 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.

Part II - Additional Information (This part will not be element of the MoU)

Part II-A . LIST OF EXPERTS

Expert 1.
Prof. Roderick BLOEM,Institute for Applied Information Processing and Communications, TU Graz (AT)
roderick.bloem@iaik.tugraz.at
Contacted:Yes - Possible MC:Yes
Inffeldgasse 16a, A-8010, Graz, Austria

Expert 2.
Prof. Maria Paola BONACINA,Department of Computer Science, Università degli Studi di Verona (IT)
mariapaola.bonacina@univr.it
Contacted:Yes - Possible MC:Yes
Dipartimento di Informatica Università degli Studi di Verona Ca' Vignal 2 Strada Le Grazie 15 I-37134 Verona, Italy

Expert 3.
Prof. Ahmed BOUAJJANI,University of Paris 7 (FR)
abou@liafa.jussieu.fr
Contacted:Yes - Possible MC:No

Expert 4.
Dr. Enric Rodríguez CARBONELL,Software Department, Technical University of Catalonia (UPC) (ES)
erodri@lsi.upc.edu
Contacted:Yes - Possible MC:Yes
Technical University of Catalonia (UPC) Software Department, Building Omega, Office 114, Jordi Girona, 1-3, 08034 Barcelona, Spain

Expert 5.
Prof. Silvia GHILEZAN,Faculty of Engineering, University of Novi Sad (RS)
gsilvia@uns.ns.ac.yu
Contacted:Yes - Possible MC:Yes
office 608, Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia

Expert 6.
Prof. Ian HORROCKS,Oxford University Computing Laboratory (UK)
ian.horrocks@comlab.ox.ac.uk
Contacted:Yes - Possible MC:Yes
Wolfson Building, Parks Road, Oxford OX1 3QD, UK

Expert 7.
Prof. Paul JACKSON,School of Informatics at the University of Edinburgh (UK)

pbj@inf.ed.ac.uk
Contacted:Yes - Possible MC:Yes
Room 4.05, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK

Expert 8.
Prof. Predrag JANICIC,Faculty of Mathematics, University of Belgrade (RS)
janicic@matf.bg.ac.yu
Contacted:Yes - Possible MC:Yes
Faculty of Mathematics, Studentski trg 16, 11000 Belgrade, SERBIA

Expert 9.
Prof. Viktor KUNCAK,School of Computer and Communication Sciences, Swiss Federal Institute of Technology (EPFL) (CH)
viktor.kuncak@epfl.ch
Contacted:Yes - Possible MC:Yes
EPFL / I&C / LARA / INR318 Station 14, CH-1015 Lausanne, Switzerland

Expert 10.
Prof. Marius MINEA,Department of Computing, Politehnica University of Timisoara (RO)
marius@cs.upt.ro
Contacted:Yes - Possible MC:Yes
Bd. V. Parvan nr. 2 RO-300223 Timisoara, ROMANIA

Expert 11.
Prof. Robert NIEUWENHUIS,Computer Science Department, Computer Science School (ES)
roberto@lsi.upc.edu
Contacted:Yes - Possible MC:No

Expert 12.
Prof. Tobias NIPKOW,Institut für Informatik, Technische Universität München (DE)
nipkow@in.tum.de
Contacted:Yes - Possible MC:Yes
Institut für Informatik, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany

Expert 13.
Prof. Alexander RABINOVICH,School of Computer Science, Tel Aviv University (IL)
rabinoa@post.tau.ac.il
Contacted:No - Possible MC:No
Ramat Aviv, Tel Aviv 69978, Israel

Expert 14.
Dr. Silvio RANISE,LORIA and INRIA-Lorraine (FR)
ranise@loria.fr
Contacted:Yes - Possible MC:Yes

LORIA and INRIA-Lorraine, 615, rue du Jardin Botanique, BP 101, 54602 Villers-les-Nancy, France

Expert 15.
Dr. Stefan RATSCHAN,Institute of Computer Science, Academy of Sciences of the Czech Republic (CZ)
stefan.ratschan@cs.cas.cz
Contacted:Yes - Possible MC:Yes
Pod Vodarenskou vezi 2, 182 07 Prague 8, Czech Republic

Expert 16.
Dr. Andrey RYBALCHENKO,Max Planck Institute for Software Systems (DE)
rybal@mpi-sws.mpg.de
Contacted:Yes - Possible MC:Yes
Max Planck Institute for Software Systems, Campus E1 4, D-66123 Saarbrücken, Germany

Expert 17.
Prof. Shmuel SAGIV,School of Computer Science, Tel-Aviv University (IL)
msagiv@acm.org
Contacted:Yes - Possible MC:No

Expert 18.
Prof. Natasha SHARYGINA,Informatics Department, University of Lugano (CH)
natasha.sharygina@unisi.ch
Contacted:Yes - Possible MC:Yes
Via G. Buffi 13, CH-6900 Lugano, Switzerland

Expert 19.
Dr. Tayssir TOUILLI,CNRS, Laboratoire d'Informatique Algorithmique: Fondements et Applications (FR)
touili@liafa.jussieu.fr
Contacted:Yes - Possible MC:Yes
6 th floor, Office 6A7, 175 rue du Chevaleret, F-75013 Paris, France

Part II-B. ADDITIONAL INFORMATION

HISTORY OF THE PROPOSAL

The desire for unified approaches in automated reasoning has been recognized and pursued for years by the experts interested in the Action, resulting in several discussions at scientific conferences. Discussions about coordinated activities started through a number of individual visits and meetings between the experts in years 2007 and 2008. The discussions were particularly intensified after realizing that COST is an ideal instrument to coordinate these activities and that an organized effort is necessary to ensure strong results that will have a large impact on the community and the society. These observations led the experts to jointly draft the proposal.

RECENT PUBLICATIONS BY EXPERTS

Action experts maintain an active publication record, with over 150 relevant publications in years 2007 and 2008 written by the experts that participated in preparation of the proposal. The following is a sample of 80 publications in years 2007 and 2008.

1. Ahmed Bouajjani, Anca Muscholl, Tayssir Touili: Permutation rewriting and algorithmic verification. Inf. Comput. 205(2): 199-224 (2007)
2. Ahmed Bouajjani, Séverine Fratani, Shaz Qadeer: Context-Bounded Analysis of Multithreaded Programs with Dynamic Linked Structures. CAV 2007: 207-220
3. Ahmed Bouajjani, Yan Jurski, Mihaela Sighireanu: A Generic Framework for Reasoning About Dynamic Networks of Infinite-State Processes. TACAS 2007: 690-705
4. Akash Lal, Tayssir Touili, Nicholas Kidd, Thomas W. Reps: Interprocedural Analysis of Concurrent Programs Under a Context Bound. TACAS 2008: 282-298
5. Alexander Malkis, Andreas Podelski, Andrey Rybalchenko: Precise Thread-Modular Verification. SAS 2007: 218-232
6. Alexey Gotsman, Josh Berdine, Byron Cook, Mooly Sagiv: Thread-modular shape analysis. PLDI 2007: 266-277
7. Amine Chaieb, Tobias Nipkow: Proof Synthesis and Reflection for Linear Arithmetic. J. Autom. Reasoning 41(1): 33-59 (2008)
8. Andreas Griesmayer, Stefan Staber, Roderick Bloem: Automated Fault Localization for C Programs. Electr. Notes Theor. Comput. Sci. 174(4): 95-111 (2007)
9. Andreas Podelski, Andrey Rybalchenko, Thomas Wies: Heap Assumptions on Demand. CAV 2008: 314-327
10. Andreas Podelski, Andrey Rybalchenko: Transition predicate abstraction and fair termination. ACM Trans. Program. Lang. Syst. 29(3): (2007)
11. Andrey Rybalchenko, Viorica Sofronie-Stokkermans: Constraint Solving for Interpolation. VMCAI 2007: 346-362
12. Ashutosh Gupta, Thomas A. Henzinger, Rupak Majumdar, Andrey Rybalchenko, Ru-Gang Xu: Proving non-termination. POPL 2008: 147-158
13. Barbara Jobstmann, Stefan Galler, Martin Weiglhofer, Roderick Bloem: Anzu: A Tool for Property Synthesis. CAV 2007: 258-262
14. Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter F. Patel-Schneider, Ulrike Sattler: OWL 2: The next step for OWL. J. Web Sem. 6(4): 309-322 (2008)
15. Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, Ulrike Sattler: A Logical Framework for Modularity of Ontologies. IJCAI 2007: 298-303
16. Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, Ulrike Sattler: Just the right amount: extracting modules from ontologies. WWW 2007: 717-726
17. Birte Glimm, Ian Horrocks, Carsten Lutz, Ulrike Sattler: Conjunctive Query Answering for the Description Logic SHIQ. IJCAI 2007: 399-404
18. Boris Motik, Ian Horrocks, Ulrike Sattler: Bridging the gap between OWL and relational databases. WWW 2007: 807-816
19. Boris Motik, Ian Horrocks: OWL Datatypes: Design and Implementation. International Semantic Web Conference 2008: 307-322
20. Boris Motik, Rob Shearer, Ian Horrocks: Optimized Reasoning in Description Logics Using Hypertableaux. CADE 2007: 67-83
21. Bruno Marnette, Viktor Kuncak, and Martin Rinard. Polynomial constraints for sets with cardinality bounds. In Foundations of Software Science and Computation Structures (FOSSACS), volume 4423 of LNCS, March 2007.
22. Byron Cook, Alexey Gotsman, Andreas Podelski, Andrey Rybalchenko, Moshe Y. Vardi: Proving that programs eventually do something good. POPL 2007: 265-276

23. Byron Cook, Andreas Podelski, Andrey Rybalchenko: Proving thread termination. PLDI 2007: 320-330
24. Byron Cook, Daniel Kroening, Natasha Sharygina: Verification of Boolean programs with unbounded thread creation. Theor. Comput. Sci. 388(1-3): 227-242 (2007)
25. Byron Cook, Sumit Gulwani, Tal Lev-Ami, Andrey Rybalchenko, Mooly Sagiv: Proving Conditional Termination. CAV 2008: 328-340
26. Chiara Braghin, Natasha Sharygina, Katerina Barone-Adesi: Automated Verification of Security Policies in Mobile Code. IFM 2007: 37-53
27. Daniel Kroening, Natasha Sharygina, Stefano Tonetta, Aliaksei Tsitovich, Christoph M. Wintersteiger: Loop Summarization Using Abstract Transformers. ATVA 2008: 111-125
28. David Déharbe, Silvio Ranise, Jorgiano Vidal: Distributing the Workload in a Lazy Theorem-Prover. Electr. Notes Theor. Comput. Sci. 184: 21-37 (2007)
29. Dirk Beyer, Thomas A. Henzinger, Rupak Majumdar, Andrey Rybalchenko: Invariant Synthesis for Combined Theories. VMCAI 2007: 378-394
30. Dirk Beyer, Thomas A. Henzinger, Rupak Majumdar, Andrey Rybalchenko: Path invariants. PLDI 2007: 300-309
31. Dmitry Tsarkov, Ian Horrocks, Peter F. Patel-Schneider: Optimizing Terminological Reasoning for Expressive Description Logics. J. Autom. Reasoning 39(3): 277-316 (2007)
32. Enric Rodríguez-Carbonell, Deepak Kapur: Generating all polynomial invariants in simple loops. J. Symb. Comput. 42(4): 443-476 (2007)
33. Felix Klaedtke, Stefan Ratschan, Zhikun She: Language-Based Abstraction Refinement for Hybrid System Verification. VMCAI 2007: 151-166
34. Gaël Patin, Mihaela Sighireanu, Tayssir Touili: Spade: Verification of Multithreaded Dynamic and Recursive Programs. CAV 2007: 254-257
35. Germain Faure, Robert Nieuwenhuis, Albert Oliveras, Enric Rodríguez-Carbonell: SAT Modulo the Theory of Linear Arithmetic: Exact, Inexact and Commercial Solvers. SAT 2008: 77-90
36. Giorgos Stoilos, Giorgos B. Stamou, Jeff Z. Pan, Vassilis Tzouvaras, Ian Horrocks: Reasoning with Very Expressive Fuzzy Description Logics. J. Artif. Intell. Res. (JAIR) 30: 273-320 (2007)
37. Greta Yorsh, Thomas W. Reps, Mooly Sagiv, Reinhard Wilhelm: Logical characterizations of heap abstractions. ACM Trans. Comput. Log. 8(1): (2007)
38. Guy Gueta, Cormac Flanagan, Eran Yahav, Mooly Sagiv: Cartesian Partial-Order Reduction. SPIN 2007: 95-112
39. Görschwin Fey, Stefan Staber, Roderick Bloem, Rolf Drechsler: Automatic Fault Localization for Property Checking. IEEE Trans. on CAD of Integrated Circuits and Systems 27(6): 1138-1149 (2008)
40. Himanshu Jain, Daniel Kroening, Natasha Sharygina, Edmund M. Clarke: Word-Level Predicate-Abstraction and Refinement Techniques for Verifying RTL Verilog. IEEE Trans. on CAD of Integrated Circuits and Systems 27(2): 366-379 (2008)
41. Hugo Herbelin, Silvia Ghilezan: An approach to call-by-name delimited continuations. POPL 2008: 383-394
42. Ian Horrocks, Ulrike Sattler: A Tableau Decision Procedure for SHOIQ. J. Autom. Reasoning 39(3): 249-276 (2007)
43. Ian Horrocks: Logic for Ontology Engineering Corner. J. Log. Comput. 17(4): 615 (2007)
44. Ian Horrocks: Ontologies and the semantic web. Commun. ACM 51(12): 58-67 (2008)
45. Jeff Z. Pan, Ian Horrocks: RDFS(FA): Connecting RDF(S) and OWL DL. IEEE Trans. Knowl. Data Eng. 19(2): 192-206 (2007)
46. Josh Berdine, Tal Lev-Ami, Roman Manevich, G. Ramalingam, Shmuel Sagiv: Thread Quantification for Concurrent Shape Analysis. CAV 2008: 399-413
47. Karen Zee, Viktor Kuncak, and Martin Rinard. Full functional verification of linked data structures. PLDI, 2008.
48. Karin Greimel, Roderick Bloem, Barbara Jobstmann, Moshe Y. Vardi: Open Implication. ICALP (2) 2008: 361-372

49. Makarius Wenzel, Lawrence C. Paulson, Tobias Nipkow: The Isabelle Framework. TPHOLs 2008: 33-38
50. Maria Paola Bonacina, Mnacho Echenim: On Variable-inactivity and Polynomial T-Satisfiability Procedures. J. Log. Comput. 18(1): 77-96 (2008)
51. Maria Paola Bonacina, Mnacho Echenim: T-Decision by Decomposition. CADE 2007: 199-214
52. Maria Paola Bonacina, Nachum Dershowitz: Abstract canonical inference. ACM Trans. Comput. Log. 8(1): (2007)
53. Maria Paola Bonacina, Nachum Dershowitz: Canonical Inference for Implicational Systems. IJCAR 2008: 380-395
54. Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, Jovanka Pantovic, Daniele Varacca: Security types for dynamic web data. Theor. Comput. Sci. 402(2-3): 156-171 (2008)
55. Milena Vujosevic-Janicic, Jelena Tomasevic, Predrag Janicic: Random k-GD-SAT Model and its Phase Transition, Journal of Universal Computer Science, Vol. 13, No. 4, pp. 572-591. 2007.
56. Miquel Bofill, Robert Nieuwenhuis, Albert Oliveras, Enric Rodríguez-Carbonell, Albert Rubio: The Barcelogic SMT Solver. CAV 2008: 294-298
57. Mohamed Faouzi Atig, Ahmed Bouajjani, Tayssir Touili: On the Reachability Analysis of Acyclic Networks of Pushdown Systems. CONCUR 2008: 356-371
58. Noam Rinetzky, G. Ramalingam, Shmuel Sagiv, Eran Yahav: On the complexity of partially-flow-sensitive alias analysis. ACM Trans. Program. Lang. Syst. 30(3): (2008)
59. Paul B. Jackson, Bill J. Ellis and Kathleen Sharp: Using SMT Solvers to Verify High-Integrity Programs. 2nd International Workshop on Automated Formal Methods, (AFM '07). Atlanta, Georgia, USA, November 2007.
60. Parosh Aziz Abdulla, Ahmed Bouajjani, Lukás Holík, Lisa Kaati, Tomás Vojnar: Computing Simulations over Tree Automata. TACAS 2008: 93-108
61. Peter F. Patel-Schneider, Ian Horrocks: A comparison of two modelling paradigms in the Semantic Web. J. Web Sem. 5(4): 240-250 (2007)
62. Predrag Janicic and Alan Bundy. Automatic Synthesis of Decision Procedures: a Case Study of Ground and Linear Arithmetic, Kauers et al. (Eds.) Towards Mechanized Mathematical Assistants, Lecture Notes in Computer Science, 4573, pp. 80–93. Springer-Verlag, Berlin-Heidelberg, 2007.
63. Predrag Janicic, Pedro Quaresma: Automatic Verification of Regular Constructions in Dynamic Geometry Systems, Francisco Botana and Tomas Recio (Eds.) Automated Deduction in Geometry, Lecture Notes in Artificial Intelligence, 4869, Springer-Verlag, Berlin-Heidelberg, 2007.
64. Robert Nieuwenhuis, Albert Oliveras: Fast congruence closure and extensions. Inf. Comput. 205(4): 557-580 (2007)
65. Roderick Bloem, Roberto Cavada, Ingo Pill, Marco Roveri, Andrei Tchaltsev: RAT: A Tool for the Formal Analysis of Requirements. CAV 2007: 263-267
66. Roman Manevich, Josh Berdine, Byron Cook, G. Ramalingam, Mooly Sagiv: Shape Analysis by Graph Decomposition. TACAS 2007: 3-18
67. Ruzica Piskac and Viktor Kuncak. Decision procedures for multisets with cardinality constraints. In 9th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), LNCS, 2008.
68. Ruzica Piskac and Viktor Kuncak. Linear arithmetic with stars. In Computed-Aided Verification (CAV), volume 5123 of LNCS, 2008.
69. Sagar Chaki, Edmund M. Clarke, Natasha Sharygina, Nishant Sinha: Verification of evolving software via component substitutability analysis. Formal Methods in System Design 32(3): 235-266 (2008)
70. Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, Daniele Zucchelli: Combination Methods for Satisfiability and Model-Checking of Infinite-State Systems. CADE 2007: 362-378
71. Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, Daniele Zucchelli: Decision procedures for extensions of the theory of arrays. Ann. Math. Artif. Intell. 50(3-4): 231-254 (2007)
72. Silvio Ghilardi, Enrica Nicolini, Silvio Ranise, Daniele Zucchelli: Towards SMT Model Checking of

Array-Based Systems. IJCAR 2008: 67-82
73. Silvio Ranise, Christophe Ringeissen, Duc-Khanh Tran: Combining Proof-Producing Decision Procedures. FroCos 2007: 237-251
74. Stefan Ratschan, Zhikun She: Safety verification of hybrid systems by constraint propagation-based abstraction refinement. ACM Trans. Embedded Comput. Syst. 6(1): (2007)
75. Stefan Staber, Roderick Bloem: Fault Localization and Correction with QBF. SAT 2007: 355-368
76. Thanh Tran, Peter Haase, Boris Motik, Bernardo Cuenca Grau, Ian Horrocks: Metalevel Information in Ontology-Based Applications. AAAI 2008: 1237-1242
77. Tobias Nipkow: Linear Quantifier Elimination. IJCAR 2008: 18-33
78. Viktor Kuncak and Martin Rinard. Towards efficient satisfiability checking for Boolean Algebra with Presburger Arithmetic. In Conference on Automateded Deduction (CADE-21), volume 4603 of LNCS, 2007.
79. Werner Damm, Guilherme Pinto, Stefan Ratschan: Guaranteed Termination in the Verification of LTL Properties of Non-linear Robust Discrete Time Hybrid Systems. Int. J. Found. Comput. Sci. 18(1): 63-86 (2007)
80. Thomas Wies, Viktor Kuncak, Karen Zee, Andreas Podelski, and Martin Rinard. Verifying complex properties using symbolic shape analysis. In *Workshop on Heap Abstraction and Verification*, 2007.

EARLY STAGE RESEARCHERS

The best way to promote and train early stage researchers such as PhD students is to involve them into activities of multiple research groups. This Action therefore proposes to minimize other expenses (such as additional yearly meetings) and instead focus its resources on short-term scientific missions for PhD students.

BROAD PARTICIPATION AND GENDER BALANCE

The Action is doing all that is in its power to attract the top researchers from all eligible countries and from all broadly relevant areas. The Action is also committed to promoting gender balance, as described in Section E.4. If approved, the Action membership is expected to grow further both in terms of the number of researchers and the number of participating countries. The program provisions for this growth by the broad nature of the proposed activity and by provisioning for up to two more work groups.

MONITORING PROVISIONS

To ensure progress towards objectives, work group coordinators will monitor the evaluation of objectives and milestones and summarize current progress at yearly meetings and reports.

INDUSTRIAL APPLICATIONS

The activities will continuously be based on practical applications. These applications will generate benchmarks suite of Work Group 1, which will be the driving force of the Rich Model Toolkit and the entire Action. The Action will specifically invite members of major research labs and companies with large research and development efforts to explore the potential for technology transfer and the building of industrial tools based on Rich-Model Toolkit. Specific industry contacts include researchers in:

1. IBM Zurich
2. IBM Haifa
3. Intel Haifa
4. ABB
5. ST Microelectronics Grenoble

6. Infineon
7. OneSpin
8. Nokia
9. Praxis High Integrity Systems

A SAMPLE OF NATIONAL PROJECTS COORDINATED BY THE ACTION

The following is a sample list of national projects (sorted by the country) in which Action experts are currently involved.

1. Germany: Verisoft (http://www.verisoft.de/)
2. Germany: AVACS (http://www.avacs.org/)
3. Germany: Ph.D. programme Puma (http://puma.in.tum.de/)
4. UK: EP/F065841/1 HermiT: Reasoning with Large Ontologies (http://gow.epsrc.ac.uk/ViewGrant.aspx?GrantRef=EP/F065841/1)
5. UK: EP/C543319/2 LOGO: Logics for Ontologies (http://gow.epsrc.ac.uk/ViewGrant.aspx?GrantRef=EP/C543319/2)
6. UK: EP/E03781X/1 Reasoning Infrastructure for Ontologies and Instances (http://gow.epsrc.ac.uk/ViewGrant.aspx?GrantRef=EP/E03781X/1)
7. UK: EP/C537211/2 REOL: Reasoning for Expressive Ontology Languages (http://gow.epsrc.ac.uk/ViewGrant.aspx?GrantRef=EP/C537211/2)
8. Spain:LogicTools Project, funded by the Spanish Spanish Ministry of Science and Education
9. Serbia: National Project 144030 funded by the Serbian Ministry of Science
10. Serbia: PROJECT 144029, Models, Languages, Types, and Processes in Computing funded by the Serbian Ministry of Science
11. Switzerland: Detection of Security Flaws and Vulnerabilities by Guided Model Checking, Swiss National Science Foundation.
12. Switzerland: Formal Verification Techniques for Security. Tasso Foundation
13. Switzerland: Precise and Scalable Analyses for Reliable Software, Swiss National Science Foundation.
14. Italy: Integrating automated reasoning in model checking: towards push-button formal verification of large-scale and infinite-state systems.
15. Romania: CONQUERS: Continuous Quality Evaluation and Restructuring of Software (Romanian national research grant, 2007-2010)
16. Czech science foundation project GACR 201/08/J020 “Verification of Hybrid Systems - Exploiting the Synergy with Underlying Constraint Solving Technology”

The above projects are among the projects whose activities will be coordinated by the Action. The experts anticipate initiation of additional projects in the course of the Action to complement currently running programs by funding for additional concrete research tasks. The present Action will prevent the duplication of effort in the newly proposed projects.