Expression of interest for a Network of Excellence on Abstract Interpretation (AINoE)

The information society technologies are one of the most important sectors of the European economy. The aim of the network is to mobilise the abstract interpretation community that is scattered across Europe, unify it and galvanise it into providing technologies for the production of composable, reliable, scalable and robust software. The network will provide a route for Europe to benefit from this technology which itself originates from Europe.

1 Need and Relevance

The network of excellence aims at strengthening European scientific and technological excellence through a better integration of research capacities on abstract interpretation and its applications across Europe. Software and hardware reliability, trustworthiness and security is a major societal and economic challenge for information technology based modern societies. Many semantic based program analysis and manipulation techniques have been developed in the past 20 years offering more rigour than testing and simulation and greater scalability than verification by automatic theorem proving. This includes research areas such as static analysis, abstract model checking, type and effect systems, program transformation, etc. where European researchers are leading the World. There is a profound unity in all these fields which are closely related since they can all be formalized in terms of abstract interpretation. They all aim at obtaining safe but approximate information about the behaviour of a computer system by systematically abstracting the semantics of its specification, model or program.

The network will help to integrate the various sub-communities which are centred around more specific types of analysis problems as posed by specific hardware and software specification and programming paradigms (imperative, functional, constraint/logic, object-oriented) or by particular applications (composable, scalable and reliable abstract interpretation-based software technologies for security, distribution, mobility, real-time embedded systems, knowledge bases, etc.). A closer interaction between the different groups and directions will lead to a critical mass with a huge cooperation pay-off leading to fruitful cross fertilization and new interactions. This will also intensify research by more rapidly spreading new results, reducing time to market of new ideas and supporting the currently rapidly expanding industrialization phase.

Abstract interpretation is now at a pivotal moment since it is on the verge of widespread industrialisation, and therefore the network will act to spread European scientific excellence which will, in turn, feed the industrial sector. Exchanges between academic researchers and end-users will focus research on practical issues. The ”technology-push practice-pull” organisation of the network will accelerate the maturation and the industrialisation of abstract interpretation based technologies. This expression of interest is supported by AbsInt Angewandte Informatik GmbH, Airbus France S.A.S., Carlstedt Research & Technology AB, Exalead S.A., List Group S.p.A., METAFrame Technologies and Oakwood Computing Associates Ltd..

1.1 Proposed activities

1.1.1 Research activities

Semantic foundations: Abstract interpretation based methods rely on sound abstractions/approximations of the semantics of computer systems. The formalization of such semantics, at various levels of detail, is therefore fundamental. The work ranges from geometric semantics of parallelism and distributed systems (to avoid combinatorial explosions in the analysis of interacting processes), to semantics for mobile processes or the semantic definition of query languages for the web.

Mathematical foundations and abstract domains: The central idea in abstract interpretation is to specify/compute information about computer systems by systematic abstractions/approximations of their semantics to construct simplified/effective abstract semantics/models. Abstract domains are mathematical/algebraic models specifying which properties of the computer system are considered relevant for its analysis/transformation or synthesis from specifications. Methodologies for semantics approximation and systematic design of abstract domains are required to cope with the wide variety of paradigms (imperative, functional, declarative, object oriented, parallel including reactive, (a)synchronous, distributed, mobile), properties (correctness, security, performance) and applications to be considered. Examples include spatial approximations (such as grammar-based abstract domains combined with constraints), qualitative approximations (such as probabilistic abstract domains), generalizations of domain-specific analysis techniques, resulting in a much more useful technique, etc. Important aspects for both the precision and efficiency of analyses/transformations include compositionality (to cope with interactions of analyses/transformations), modularity (e.g. by taking advantage of new mechanisms for program structuring such as modules or aspects, or by developing new, incremental approaches to program certification, focussing on parts of possibly shared code, while disregarding the rest), formal assessment of precision, automated refinement of abstraction, etc.

Algorithmic and generic implementation foundations: The abstract semantics/models are generally expressed in equivalent computer-representable fixpoint/constraint forms. Efficient iterative resolution/constraint solving algorithms constitute the kernel of generic static analysis and transformation techniques. The algorithmic core of most program analysis/transformation systems is the computation of a (preferably least) solution to a system of constraints. Scalability of the system therefore crucially depends on efficient fixpoint/reasoning engines for computing such solutions. These engines will equip the community with tools to develop further analyses. Practical program analysis tools should be easily adaptable to newly arising analysis questions. Adaptivity usually is gained by a general framework and analyser generators which then only must be instantiated for the concrete analysis.

Domain-specific static analysis and transformation techniques: Analysis and transformation frameworks must be developed and implemented for various programming paradigms (in a uniform way covering e.g. procedural languages (including logic languages as well as imperative languages), multi-threaded parallel code, various distributed/mobile calculi (including Mobile Ambients and Spi), reactive embedded systems (either synchronous or asynchronous), etc.).

Another dimension in the design of abstract domains is the type of properties to be considered, ranging from *precision properties* (for reasoning about the accuracy of floating point computations), to *spatial properties* (e.g. for reasoning about arbitrary large sets of mutable data structures or control flow analyses of mobile processes/ambients calculi at high level or Java/scripting languages supporting wireless technologies at low level), *temporal/performance properties* (complexity/termination/worst case execution time analysis/debugging, analysis of discrete/discretized/continuous/hybrid systems) and *security properties* (for trust and confidence analysis of programs). For each class of property, a wide range of abstract domains has to be explored with different cost/precision ratio to best fit to the considered application (e.g. precise analysis of a program component versus rapid analysis of huge programs).

Applied Software Technologies: The work on static analysis and transformation will be done in the context of various applied software technologies including classical program optimisation (in particular for functional/constraint/logical programming languages), static debugging in pervasive systems (e.g. mobile, wearable systems, embedded system-on-chip), program debugging (to help automate test generation for bug detection) and program synthesis (in particular program specialisation and partial evaluation). In particular the network will encourage the transfer of successful technologies between areas (e.g. the application of tools for CLP to other languages and logics (such as Java, Java bytecode, Verilog, first order logic) using explicit semantics expressed as CLP programs).

Abstract interpretation also provides underpinning technology for developing new software technologies such as aiding program understanding (to support the programmer in the task of reasoning about code developed by a third party), the automated certification of communication protocols/ distributed programs for network safety and data security purposes (e.g. intrusion detection for smartcards or internet-based systems to automatically detect inconspicuous program features that elite hackers can exploit and potentially lever into major security holes), the optimisation of data mining tools implemented on top of a logic programming engine or tool support for production of intelligent, knowledge-based systems which are presently lacking.

1.2 Relevant priority thematic area

The proposed activities aim at fostering research and development of a wide spectrum of techniques based on abstract interpretation by *integrating and strengthening the European research area* (1.). The considered application areas of abstract interpretation based program analysis and manipulation techniques underlie many *Information Society technologies* (1.1.2) including *major societal and economic challenges* (1.1.2.i) such as *technologies for trust and security* [e.g. the verification of cryptographic protocols, confidentiality analysis, semantic program watermarking in dynamic and mobile systems], as well as *communication, computing and software technologies* (1.1.2.ii) in particular *software technologies, embedded systems and distributed systems* for which the consortium proposes to develop new software technologies for software and systems that address composability, scalability, reliability and robustness. The domain of application is that of the formal design and verification of computer systems including complex embedded distributed systems [in particular as found in *aeronautics* (1.1.4.i)] as well as *components and microsystems* (1.1.2.iii) [such as systems-on-a-chip], *knowledge and interface technologies* (1.1.2.iv) [such as semantic-web applications], *nanotechnologies* (1.1.3) [probabilistic verification], etc. The research program should foster the industrialization of abstract interpretation based software technology as an *IST future and emerging technology* (1.1.2.v).

1.3 Mobilisation of activities and resources

1.3.1 Application Domains

The main application areas will be defined according to the end-user needs, in particular applications requiring the development of software critical technologies for key safety and security challenges posed by the "all-digital" world and by the need to secure the rights of individuals and communities. This includes:

aeronautics and space: by developing tools for detection of programming errors to reduce test costs and enhance reliability of software (ranging from safety-sensitive critical control functions up to large high-availability embedded distributed data processing);

communication, computing, software technologies for distributed systems: by using abstract interpretation to explore designs, prototypes and implementations for matching with specifications for all the software lifetime (from first development, including the corrective and evolutive maintenance for tens of years or over);

technology for trust and security: by using abstract interpretation to track information flow in protocols and systems;

multifunctional service creation environments: by exploiting ideas from abstract interpretation and model checking as well as model synthesis technologies, in order to support the development, increase the reliability and guarantee quality of service.

1.3.2 Education activities

The network will promote scientific education and help establishing software reliability as a fundamental topic in the computer science curriculum. The members of the network are both World specialists in their area and committed researchers and mentors. The network will promote courses for students given by group members at other member institutions (including the Dagstuhl activities), summer schools, the exchange of students, and the distribution of course materials. The PhD output of the participants will be in peak shape to join the matching industries.

2 Excellence

The expression of interest for a network of excellence on abstract interpretation is at the initiative of the following participants (given with hyperlinks to their institution, ✉ (postal address), 📧 (email), 📄 (curriculum vitæ), 🏠 (home page), 👥 (group and size, i.e. PhDs and non-PhDs with at least 4 years of experience), 📚 (list of publications), 📑 (DBLP bibliography) and ☉ (European projects)).

François Bourdoncle, Exalead S.A. (France) ✉ 📧 📄 🏠 👥 (3) 📚 📑
Maurice Bruynooghe, Katholieke Universiteit Leuven (Belgium) ✉ 📧 📄 🏠 👥 (6) 📚 📑 ☉
Michael Codish, Ben-Gurion University of the Negev (Israel) ✉ 📧 📄 🏠 👥 (3) 📚 📑 ☉
Agostino Cortesi, Universit Ca' Foscari di Venezia (Italy) ✉ 📧 📄 🏠 👥 (10) 📚 📑
Patrick Cousot, École Normale Supérieure/CNRS (France) ✉ 📧 📄 🏠 👥 (4) 📚 📑 ☉
Radhia Cousot, École Polytechnique/CNRS (France) ✉ 📧 📄 🏠 👥 (2) 📚 📑 ☉
Olivier Danvy, University of Aarhus (Denmark) ✉ 📧 📄 🏠 👥 (5) 📚 📑 ☉
Javier Esparza, University of Edinburgh (UK) ✉ 📧 📄 🏠 👥 (4) 📚 📑 ☉
Christian Ferdinand, AbsInt Angewandte Informatik GmbH (Germany) ✉ 📧 🏠 👥 (5) 📚 📑
Gilberto Filé, Universitá di Padova (Italy) ✉ 📧 📄 🏠 👥 (4) 📚 📑 ☉
John Gallagher, Roskilde University (Denmark) ✉ 📧 📄 🏠 👥 (1) 📚 📑 ☉
Roberto Giacobazzi, Universitá degli Studi di Verona (Italy) ✉ 📧 📄 🏠 👥 (4) 📚 📑 ☉
Éric Goubault, CEA/LIST (France) ✉ 📧 🏠 👥 (3) 📚 📑 ☉
Nicolas Halbwachs, Vérimag/CNRS (France) ✉ 📧 📄 🏠 👥 (5) 📚 📑 ☉
Chris Hankin, Imperial College (UK) ✉ 📧 📄 🏠 👥 (5) 📑 ☉
Manuel Hermenegildo, Universidad Politécnica de Madrid (Spain) ✉ 📧 📄 🏠 👥 (7) 📚 📑 ☉
John Hughes, University of Chalmers (Sweden) ✉ 📧 📄 🏠 👥 (1) 📚 📑 ☉
Thomas Jensen, IRISA/CNRS (France) ✉ 📧 🏠 👥 (5) 📚 📑
Neil Jones, DIKU, University of Copenhagen (Denmark) ✉ 📧 📄 🏠 👥 (8) 📚 📑 ☉
Andy King, University of Kent (UK) ✉ 📧 📄 🏠 👥 (3) 📚 📑 ☉
Baudouin Le Charlier, Université Catholique de Louvain (Belgium) ✉ 📧 📄 🏠 👥 (1) 📚 📑 ☉
Giorgio Levi, Universitá di Pisa (Italy) ✉ 📧 📄 🏠 👥 (5) 📚 📑 ☉
Alan Mycroft, University of Cambridge (UK) ✉ 📧 📄 🏠 👥 (3) 📚 📑
Flemming Nielson, The Technical University of Denmark (Denmark) ✉ 📧 📄 🏠 👥 (2) 📚 📑 ☉
Andreas Podelski, Max-Planck-Institut für Informatik (Germany) ✉ 📧 📄 🏠 👥 (2) 📚 📑 ☉
German Puebla, Universidad Politécnica de Madrid (Spain) ✉ 📧 📄 🏠 👥 (7) 📚 📑 ☉
Famantanantsoa Randimbivololona, Airbus France (France) 📧 🏠 👥 (3) 📑
Hanne Riis Nielson, The Technical University of Denmark (Denmark) ✉ 📧 📄 🏠 👥 (1) 📚 📑 ☉
Mooly Sagiv, Tel-Aviv University (Israel) ✉ 📧 📄 🏠 👥 (1) 📚 📑
David Sands, Chalmers University of Technology (Sweden) ✉ 📧 📄 🏠 👥 (5) 📚 📑 ☉
Helmut Seidl, Universität Trier (Germany) ✉ 📧 📄 🏠 👥 (2) 📚 📑
Bernhard Steffen, Universität Dortmund (Germany) ✉ 📧 📄 🏠 👥 (5) 📚 📑
Reinhard Wilhelm, Universität des Saarlandes (Germany) 📧 🏠 👥 (10) 📚 📑

The proposers head research groups totalling 132 researchers (excluding PhD students). The members of the network have a strong background in the theory and application of semantics based techniques to problems of program analysis and software development. They already have a proven record of collaboration in research, organisation of joint workshops and conferences (notably ESOP, PEPM, SAS and VMCAI), courses for graduate students and contacts with industrial partners. The network will have an opened structure to welcome other European participants willing to join.

The effort should be international and therefore the participants promote a strong international cooperation in particular with Australia (Harald Søndergaard (Melbourne University), Kim Marriott (Monash University)), India (R.K. Shyamasundar (Tata Institute of Fundamental Research)), Korea (Kwangkeun Yi (KAIST)) and the USA (Alex Aiken (Berkeley University), Saumya Debray (The University of Arizona), Michael Lowry (NASA Ames Research Center), Jens Palsberg (Purdue University), Thomas Reps (University of Wisconsin-Madison), Barbara Ryder (Rutgers University), David Schmidt (Kansas State University), Pascal van Hentenryck (Brown University)) to cite a few. The international activities of the network will be through meetings at the Schloß Dagstuhl international conference and research center for computer science, the Static Analysis Symposium (SAS) and the International Conference on Verification, Model Checking and Abstract Interpretation (VMCAI).

3 Integration and Structuring Effect

Europe already plays a leading role in the development and application of static analysis techniques in novel application areas. To fully realise this potential and exploit it before other parts of the World, it is essential that these techniques are mastered by the partners and distilled into implementations and tools that are semantically correct, informative and algorithmically efficient. This will allow European Industry to develop more reliable and trustworthy software, even when based on components developed using less rigourous techniques.

The European cooperation in this area is often organized nationally or through small groups addressing separately various applications. The network will therefore widen and deepen their collaboration. There is a need to disseminate formal methods for software engineering in the European industry where the uptake is slow (though growing) and restricted to certain dedicated areas. A network visible at the European level with existing contacts to industry will be able to organize focussed tutorials and workshops and coordinate the use of techniques and tools in European projects where software analysis is being employed.

The basic integration measures propounded by the network will include e-mail list, web site, database of interests and skills, periodic meetings in the different countries participating to the network, post-doc exchanges and fellowships.

The technical integration measures proposed by the network include the development of composable libraries implementing abstract domains to be used in sharable program analysis engines.

At the educational level, the network will advocate an international PhD programme in abstract interpretation-based software technologies with a unique advisory board for the PhD, made of the representatives of the different sites. The objective is to support collaborations through common rules for student financial support, student/teaching exchange and partnership (evaluation, co-tutoring, visiting vacancies, thesis standards etc.). This program would be enhanced by a high-speed satellite connectivity among the different partners of the network. This technology is essential to share seminars, teaching and meetings in video-conferencing.

The involvement with spin-offs and start-up companies as well as the academic/industrial integration will be put forward by industrial fellowships (that is visiting positions from academia to industry and vice versa), formulation of shared course materials, training courses for industry.