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

Brussels, 14 November 2014

COST 102/14

MEMORANDUM OF UNDERSTANDING

Subject :	Memorandum of Understanding for the implementation of a European Concerted Research Action designated as COST Action IC1405: Reversible computation - extending horizons of computing

Delegations will find attached the Memorandum of Understanding for COST Action IC1405 as approved by the COST Committee of Senior Officials (CSO) at its 191th meeting on 12-13 November 2014.

MEMORANDUM OF UNDERSTANDING

For the implementation of a European Concerted Research Action designated as

COST Action IC1405
REVERSIBLE COMPUTATION: EXTENDING HORIZONS OF COMPUTING

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

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

2. The main objective of the Action is to establish a platform to coordinate European research on reversible computation, to expand and strengthen a network of European reversible computation researchers and practitioners, and to cooperate with industrial partners on evaluating and adopting solutions delivered during the Action.

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 64 million in 2014 prices.

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

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

A. ABSTRACT AND KEYWORDS

Reversible computation is an emerging paradigm that extends the standard forwards-only mode of computation with the ability to execute in reverse, so that computation can run backwards as naturally as it can go forwards. It aims to deliver novel computing devices and software, and to enhance traditional systems by equipping them with reversibility. The potential benefits include the design of revolutionary reversible logic gates and circuits - leading to low-power computing and innovative hardware for green Information and Communication Technologies (ICT), and new conceptual frameworks, language abstractions and software tools for reliable and recovery oriented distributed systems.

Landauer's Principle, a theoretical explanation why a significant proportion of electrical power consumed by current forwards-only computers is lost in the form of heat, and why making computation reversible is necessary and beneficial, has only been shown empirically in 2012. Hence now is the right time to launch a COST Action on reversible computation. The Action will establish the first European (and the world first) network of excellence to coordinate research on reversible computation. Many fundamental challenges cannot be solved currently by partitioned and uncoordinated research, so a collaborative effort of European expertise with an industrial participation, as proposed by this Action, is the most logical and efficient way to proceed.
Keywords: Physical and logical reversible computation; models of reversibility: automata and process calculi; reversible algorithms, programming languages and software architectures; reversible and quantum logics and circuits

B. BACKGROUND

B.1 General background

Computing supports and enhances all aspects of modern society's life, thus there are costs and risks of such critical dependence. The current computer technology is based on an irreversible mode of computation, hence is very energy hungry. There is also a limit to the extent miniaturisation can increase the speed and capacity of circuits. At the level of systems, modern hardware runs distributed and communication-based software. The correctness and reliability of such software is essential but occasionally programs deadlock, data is corrupted and security is breached. Despite much effort not many effective techniques exist for recovery from such failures. The ever-increasing demand for electricity, hardware limitations of current technology, and the need for robust recovery-oriented systems are only some reasons why scientists and developers have been exploring alternative computation paradigms in recent years.

Reversible computation is one such radical idea. It originates from an observation by Landauer fifty years ago that, although physical processes are reversible at a microscopic level, current CMOS (complementary metal-oxide-semiconductor) computer technology is based on irreversible execution. Circuits are built from irreversible logic gates, such as NAND (not-and) gates, which erase information as they operate, resulting in heat dissipation. Thus circuits require constant supply of electricity to power them. There is no reason, however, why computation needs to be irreversible. Reversible Turing machines and a number of reversible logic gates were developed in the 1970s and 1980s. They showed that reversible computers can be built in principle. Since such computers would not erase information, heat dissipation and thus power consumption would be decreased. There are other factors for energy loss in computers, so reversibility aims at reducing drastically power consumption rather than eliminating it. Recently, small prototype reversible circuits have been implemented in CMOS and the novel SQUID (superconducting quantum interference device) and MEMS (microelectromechanical systems) technologies. Running them adiabatically leads to a tenfold energy saving thus holding potentially the promise of solving the energy problem. Very recently, experiments with AQFP (Adiabatic Quantum-Flux-Perametron) logic gates based on superconductors have shown the reversible circuits can be created and that the energy loss is negligible.

Apart from holding the promise of solving the energy problem reversibility has found many exciting and valuable application areas in the last decade. For example, reversibility is useful in fault-tolerance, reliability analysis, static timing analysis of programs, database view-updating, simulation, modelling of biochemistry and quantum computing. The theoretical breakthroughs include a reversible imperative programming language, reversible process calculi and cellular automata, understanding of the relationship between distributed behaviour, causality and reversibility, reversible operational semantics, and time and space complexity results.

Reversible computation postulates that computing devices, systems and software can be constructed to operate with minimal information loss (thus heat dissipation) by supporting both forwards and reverse computation, and that such a mode of operation aligns naturally with many (described above) computation tasks. The central research goal of this project is to continue developing reversible computation as a promising paradigm for future energy efficient computing and as a novel, closer to the physical reality, approach to understanding, modelling, programming and verifying complex software and systems. More specifically the Action aim to

- Create theoretical foundations of reversibility; to discover its purpose and benefits in natural and artificial processes, ascertain its costs and limitations, and to explore how reversibility can help us to solve scientific and engineering open problems and challenges;

- Develop reversible object oriented and functional programming languages and software architectures for them, implement the languages, propose a theory for reversibility in distributed systems, including behavioural and logic semantics, and explore how reversibility can help in verification, testing and debugging;

- Design, synthesise and implement new reversible logics and circuits in new technologies, develop software for automatic synthesis and layout generation of reversible circuits;

- Demonstrate the benefits of reversibility in the treatment of fault-tolerance and recovery in distributed systems, long-running transactions with compensations, coding and decoding.

Scientific and engineering work on reversibility centres currently around several groups in European countries, and in the USA, Japan and Canada. These groups are typically small and there is no international coordination. In the other European countries the research on reversibility is carried out by one or two member teams and there are new groups that would like to join the reversibility community. The research itself is carried out in different contexts and at different levels of abstraction, spanning from circuits and hardware, via software to theory and formal models. As a result, it is largely independent and disconnected. Understanding the relationships between the different approaches would give a clearer picture of the whole area, and would enable cross-fertilization results. However, such an understanding can only be reached by a coordinated networking and research effort involving groups active in the different areas. Although some of the members of the community meet annually at the Reversible Computation (RC) Conference, held in Canada in 2013 and in Japan this year, there is no European network devoted to coordination of research and education of reversibility. Such a network is urgently needed to expand further, coordinate and cross-fertilise reversibility research in Europe, avoid duplication of effort, and maintain and improve Europe's position in the field, given that much significant work is taking place outside Europe. This Action will also address the urgent need for knowledge transfer between the existing and new groups, and between computer science and other disciplines, and with industry.

B.2 Current state of knowledge

A common question that runs through all reversible computation research, from reversible circuits and programming languages to reversible automata and process calculi, is what is exactly reversibility. In general, there is logical reversibility and there is physical reversibility. A computation is logically reversible if it is always possible to efficiently reconstruct the previous state of the computation from the current state. Physical reversibility means that computation can be carried out without loss of energy, or more formally, with no increase in physical entropy; and is thus energy efficient. This Action is concerned with both forms of reversibility as can be seen from its organisation into Working Groups, their research objectives and from the character of its case studies.

Computation is reversible if it can proceed in the forwards direction as well as in reverse, going back to the states visited in the past or even reaching new states that cannot be reached by going forwards alone. At a level of how computation is undone, there are three different forms of logical reversibility. *Backtracking* is when computation steps are undone in the inverse order to the order in which they have occurred, for example, backtracking in Prolog. In other words, the effects must be undone before the causes are reversed. This form of reversing ensures the property of backwards determinism: at any state in a computation, there is at most one predecessor state. In the sequential setting, whether in automata theory, sequential programming languages, or synchronous circuits and hardware, backtracking is the norm.

When computation is concurrent, distributed or asynchronous, there may be forward steps that execute independently of each other, and possibly at different locations. When such steps are reversed there is no reason to undo them in any particular order since they are independent. This gives rise to the second form of reversing called *causal order* reversing or *causal-consistent* reversing. Compensation policies of long-running transactions frequently specify reverse causal-consistent execution of compensations of transaction steps. In complex systems, for example biochemical reactions or robotics, computation steps (understood very abstractly) are undone in seemingly arbitrary order, including undoing causes before effects are undone; this is what is called *out-of-causal-order* reversing. The simplest examples concern catalysis where catalysts help molecules to bond. This form of forwards and reverse computation is common in many mechanisms in nature that aim to achieve change or progress while taking care of deadlocks and failure.

Automata and Turing machines: Logical reversibility of computing models with a finite number of discrete internal states that evolve in discrete time has been considered from different points of view. It turned out that the definitions of reversibility may have strong implications on their computational capacities and decidability properties. Models studied along these lines include the massively parallel model of cellular automata, the weakly parallel model of multi-head finite automata as well as sequential models as Turing machines, pushdown and queue automata, and finite state machines. For example, in the past few years, several different definitions of reversible multi-head reversible automata have been proposed. These have differed in just a few subtle ways, for example, one-way head motion versus two-way and reversible on all states versus reversible on reachable states. This has a dramatic impact on the computational expressiveness: one-way automata are strictly less expressive than two-way automata, and reversibility even becomes undecidable if enforced only in legal runs of the automaton. In contrast, the irreversible variants of multi-head automata are considerably more robust.

In order to understand better the sensitivity to the way reversibility is defined it is useful to study the properties of the models when different notions of reversibility are applied. In addition, to examine whether gradual reversibility increases the computational capacities can have a crucial impact on finding suitable models. For example, in complexity theory, one has the well-known space/time trade-offs of pebble games and the simulation proving that reversible space and deterministic space are equal. These results must be qualified by the exact form of reversibility used, and may not necessarily transfer to other settings with different definitions of reversibility.

A related approach is to investigate the reversibility of models that are in principle deterministic but subject to some noise. Asynchronous cellular automata are a good example. On the one hand, for cellular automata there exists well established sound basis of knowledge. On the other hand asynchronicity for cellular automata attracted quite some attention in recent years. One standard approach is to view them as automata with (some restricted kind of) probabilistic behaviour.

Programming languages: A number of interesting reversible programming languages are being actively developed. In the imperative paradigm one has Janus, and several reversible machine languages, such as PISA and BobISA. In terms of functional languages the progress is slower with RFUN, muOz and Theseus recent highlights. One of the advantages of reversible functional languages is that they support more complex data types than those of the imperative languages, which have been limited to integers, arrays and stacks.

The main challenge here is that these are still prototype languages. Thus, the code base for each of these languages is limited, and the languages do not offer many of the usual programming abstractions. This in turn has hindered the developments of reversible algorithm and useful data structures.

Research on compiler technology for reversible languages has also progressed in the last several years. One can now translate reversible high level languages to reversible machine code. The most notable developments are the Janus compiler and very recent work on how to deal with more complex heap objects. The key insight here is that translations must be clean and incur no garbage overhead. This requires novel insights and solutions which are absent in the irreversible programming languages setting.

Process calculi: When a computation is concurrent or distributed backward determinism is too restrictive, and causal reversing is the norm. The last decade has produced a good understanding of how causal reversibility can be described in the settings of operational semantics and process calculi, and how to model reversibility logically and in the terms of behavioural equivalences. Calculus of Communicating Systems (CCS) has been reversed using two different approaches (extra memory stacks for concurrent threads, and communication keys), and the reversible version of the pi-calculus has been proposed last year. Mechanisms for controlling reversibility based on a rollback construct were devised for a reversible higher-order pi-calculus, and an alternative mechanism based on the execution control operator have been also developed.

A more abstract semantic model of causal and out-of-causal order reversibility has been provided only last year in terms of modal logics and reversible event structures. Programming abstractions for out-of-causal order reversibility are also being developed.

Quantum process calculus refers to a class of formal languages in which to describe and model the behaviour of systems that combine classical and quantum communication and computation. The two main quantum process calculi are qCCS and CQP. Their development has been inspired by the success of classical process calculus as a foundation for the use of formal analysis of communication and cryptographic systems. A feature of process calculus is the development of theories of behavioural equivalence, which characterise when processes are observationally indistinguishable and can therefore be considered interchangeable. The theory and application of behavioural equivalence is the main achievement of research on quantum process calculus. The most recent work focusses on the design and implementation of automated tools for proving equivalence between given processes.

Reliable software: The idea of using reversibility for the development of reliable software is quite natural: in case of trouble one can go back to past states which were safe, and try to explore new directions avoiding the troublesome actions. Indeed, many standard techniques for reliability are based on some form of undo. For instance, checkpoints allow one to recover a past state, transactions undo their effects in case of abort, and so on.

Reversibility is also naturally applicable to debugging: when a misbehaviour is found, one can simply go back in the execution till the error causing it has been found. This is much simpler than in standard debugger, where one needs to restart the execution and put a breakpoint where one thinks the error causing it is. Indeed, many reversible debugging tools exist, and even GDB features reversible capabilities since version 7. However, to enable reversibility, one has to keep track of past history, and the related overhead (both in time and in space) may be heavy.

Nowadays, most systems are composed of many concurrent and possibly distributed entities: computers in a network, processors in a computer, cores in a processor, threads, and so on. Somehow surprisingly, a suitable definition of reversibility in a concurrent scenario, causal-consistent reversibility, has only been found in 2004. Indeed, the sequential definition of *recursively undo the last action* cannot be used, since there is no clear *last action*, but many actions executed at possibly overlapping times. According to causal-consistent reversibility, any action can be undone, provided that all the actions causally depending on it (if any) are undone beforehand. This definition relates reversibility to causality instead of time, thus allowing one to exploit it even if there is no unique notion of time for all the participants, as it may happen in distributed systems.

On top of this definition, reversible extensions of various toy concurrent languages have been developed, validating the definition and studying its consequences. In relation to reliability, formal connections between causal-consistent reversibility on one side, and transactions and rollback on the other side, have been made. Causal-consistent reversibility also enabled a new strategy for debugging concurrent systems. In fact, in concurrent systems, when replaying an execution to look for a bug, the different speed of processes may cause different behaviours, and even eliminating the misbehaviour one was trying to analyse. Using reversible debugging, one can in principle go back along the same execution, ensuring that the misbehaviour is visible. How to keep history and causality information enabling this debugging strategy is not trivial, and how to reduce the overhead is still an open problem. Recently, causal-consistent reversibility originated a new debugging strategy, where one goes back in the execution by undoing selected actions which are causes of the misbehaviour, without looking at concurrent threads of execution which are not causally related.

Control systems and robotics: Reversibility also plays a role in different programming paradigms, for example in the area of control systems and of robotics. In control systems, an interesting application concerns systems that can autonomously accumulate and revise knowledge from their own experience via self-programming, under constraints of limited time and computational resources. In this setting forward execution corresponds to deduction of the behaviour of the environment from its model, and backward execution to abduction of sub-goals from the observation of the system. Forward and backward executions of models are performed concurrently and their goodness is measured based on the quality of the predictions they make about the behaviour of the environment relative to the goals the system was designed to achieve.

For robotics, a main point is that many actions a robot performs in the real environment are reversible, e.g., changing the direction in which a mobile robot is driving, or reversing an assembly sequence of operations to perform disassembly. Thus reversibility is a useful high-level mechanism to describe sequences of operations, both for single robots and for the emergent behaviour of multi-robot swarms. Clearly, the interplay with non-reversible actions needs to be taken into account. Thus, reversibility may allow one to have for free useful reverse behaviour, or also new interesting emerging behaviours.

Reversible circuit design: Prototype reversible circuits enable several promising applications and, indeed, may be superior to conventional devices in many respects. Moreover, the inherent properties of the reversible computing paradigm can also be exploited in the design of conventional circuits and systems without changing the technology. There are numerous possible application domains of reversible circuit design; examples of four domains are given below.

Low-power circuit design may include reversible circuit components in the future. This is due to theoretical observations by Landauer stating that, independently from the applied technology, heat is always dissipated and thus power is lost when information is erased during computation. Hence, all conventional computing machines lose power as irreversible operations, such as executing the AND gate, are performed. Although the proportion of power lost due to irreversible computation is not large today, it will become substantial with the expected miniaturisation of devices and computers. Since reversible computations are information lossless, this power loss can be significantly reduced. Recently, Landauer's principle has been confirmed experimentally.

Quantum computation offers the promise of asymptotically more efficient computation of problems that are difficult for today's conventional computers. Considering that many of the established quantum algorithms include a significant Boolean component (for example, the oracle transformation in the Deutsch-Jozsa algorithm, the database in Grover's search algorithm, and the Shor's modular exponentiation algorithm), it is crucial to have efficient techniques for synthesising quantum gate realisations of Boolean functions. Since any quantum operation is inherently reversible, reversible computation can be seen as a subset of quantum computing that is easier to work with and can, thus, be exploited for this purpose.

Low-power interconnects: With the rise of very deep sub-micron and nano-metric technologies, interconnects are increasingly affecting the overall energy consumption in chips. As a result, coding strategies are applied in order to modify the communication of interconnects and, thus, reduce the power consumption. Since the applied encoders realise reversible one-to-one mappings, the application of synthesis approaches for reversible logic is a reasonable choice.

Exploitation of reversible circuit design in the *design of conventional systems* has many advantages. These include (a) the possibility to undo operations when an erroneous state has been entered, (b) the fact that reconvergences do not occur in reversible circuits (a characteristic of conventional circuits which makes important design tasks such as verification or automatic test pattern generation very complex), (c) full connectivity which makes it easy to detect an error by applying only few randomly generated stimuli, and (d) easy testability since reversible computation allows for perfect controllability and observability.

These applications provide promising perspectives to be explored and exploited. However, not all the potential of reversible circuitry has been unleashed yet. While the physicists and electrical engineers involved in the research towards these applications have made significant achievements in the recent past, there is still work to be done before this can be used for larger computing devices. However, this must not hinder research in efficient design of reversible devices. Not to slow down the process, approaches to design at the reversible logic level must be ready when the devices are.

It took over 20 years to develop an elaborated design flow composed of several abstraction levels and supported by a wide range of languages and tools for conventional circuits - still with numerous issues left unsolved. Compared to that, the design of reversible circuits is in its relative infancy. All existing approaches remain at the gate level, and no real support of reversible circuits and systems at higher levels of abstraction is available thus far. Most of the existing approaches for synthesis of circuits usually accept only specifications provided in terms of Boolean function descriptions like truth tables or Boolean decision diagrams, while only very preliminary hardware description languages are available, and they only support a very basic set of operations. At the same time, technology mapping is mainly driven by rather poor models which require a closer collaboration between the logic designers and the physicists and electrical engineers.

B.3 Reasons for the Action

Complex computer systems are a vital part of everyday life and, as the cost of their operation, maintenance and potential failure rises, there is now urgent economical and societal drive to design and eventually develop energy-conscious computing and theories and technologies for safe and efficient recovery from system failure. Reversibility offers a promising solution but, currently, only basic and experimental theories and technologies exist, and research, although concentrated and ground-breaking, is relatively fragmented and uncoordinated. Therefore, delivering sound reversibility-based theories and technologies to the computer industry and to society will require a creation of an European network to coordinate research effort, ensure flow of information and expertise between fields of research and different stakeholders, and encourage capacity building.

The main objectives of the Action will include:

- Formulating a common understanding between different fields of research what reversibility is, what purpose it serves, and how it can improve systems in order to come up with a set of challenges and the experts to solve them

- Supporting Short-Term Scientific Missions (STSM), mainly for early-career researchers and PhD students, with a view to establish common base knowledge of reversibility and to develop research collaboration, which will be supported typically at national level

- Developing Web-based knowledge and discussion forum, and organising scientific meetings and workshops

- Informing industry partners about reversibility research, cooperating on the proposed case study applications and seeking new practical applications, and setting up foundations for lasting cooperation

The networking activities of the Action will expand further, coordinate and cross-fertilise reversibility research in Europe, avoid duplication of effort, and maintain and improve Europe's position in the field.

B.4 Complementarity with other research programmes

There has been no European project that has addressed reversibility of software or hardware, or that has aimed at using theory, techniques or solutions inspired by reversible computation in ICT. This Action will be the first European (and the world first) network of excellence to coordinate, support and exploit reversible computation research.

A number of European projects aim at delivering safer, more secure and reliable software and systems, hence can be treated as sharing a proportion of aims with this COST Action. However, their approach, means and methods differ very much from ours. Three most related projects are discussed:

The EU FP7 projects HATS (Highly Adaptable and Trustworthy Software using Formal Methods, 2009 - 2013) and ENVISAGE (Engineering Virtualized Services, 2013 - 2016) place strong emphasis on the relationship between language design and the possibility of formal analysis of the resulting programs, with the first project concentrating on adaptability and evolution and the second one focusing on virtualisation and service-level agreement in the cloud. Our aim to employ reversibility at the language and tool level to achieve safer systems is similar to these projects' aims. Interaction with ENVISAGE will be granted by shared participants.

The COST Action BETTY (Behavioural Types for Reliable Large-Scale Software systems, 2012 - 2016) studies behavioural types describing the dynamic behaviour of communications inside distributed systems. Indeed, a common area of interest of the two COST Actions concerns how to exploit behavioural types to control the use of reversibility inside reversible systems, guaranteeing safety properties such as that when a participants wants to go back all the other participants are able to do so. Also, behavioural types provide a blueprint of the communication structure of the system, and can be exploited to better understand when rollback is easier or cheaper. While these questions are clearly relevant to both projects, almost no research has been done until now. Interaction with BETTY will be granted by shared participants, and at the COST general meetings.

C. OBJECTIVES AND BENEFITS

C.1 Aim

The overall goal of the Action is to establish the first European network of experts aiming at the development of reversible computation as means for safer, more reliable and recovery-oriented distributed software and systems, and, ultimately, for low-power computing.

C.2 Objectives

The prime objective of the Action is to foster and coordinate reversible computation research in Europe with a view to develop new theories, techniques and tools for safer and more reliable distributed software and systems, and to make the first steps towards creation of low-power green computing. The advances in reversible computation will improve, for example, programming of transactional systems, recovery-oriented systems and reversible debugging software. In the longer term, breakthroughs in reversible logic design and in material science will contribute significantly to the design and manufacture of reversible low-energy hardware and devices. These novel computers will run software written in reversible programming languages that this Action aims to deliver.

The Action has the following more detailed objectives:

O1. Establish a platform to coordinate and fuel European research on reversible computation with a view to

- Solve open problems and technical challenges of reversibility, and demonstrate potential benefits in computer science and in other disciplines, and in industry;
- Explore how reversibility can help in overcoming some limitations in the understanding, modelling and programming traditional software systems, and how it can enrich systems' functionality and improve reliability and security;
- Design more efficient reversible logic and quantum circuits, create prototype low-power devices; and develop new reversible programming languages and software architectures, new compilers and novel reversible algorithmics;
- Carry out case studies in order to validate the effectiveness of the reversibility-inspired theories, techniques and tools, and to point to new applications and directions of future research.

The research of the Action will be gather around four Working Groups (WG):

- WG 1 Foundations
- WG 2 Software and Systems
- WG 3 Reversible Circuit Design
- WG 4 Case Studies

O2. Expand, broaden and strengthen the network of European reversibility researchers and practitioners by organising:

- Annual Action Workshops (AWs), meeting of the Management Committee and of Working Groups
- Training schools for PhD students, early-career researchers, and scientists from other disciplines and industry

- Short-Term Scientific Missions (STSMs) between the participating groups, especially for PhD students and early-career researchers

- Peer-to-peer fact-finding and feedback visits between groups working on different aspects of reversibility and with industrial partners

- Virtual meetings to attract researchers and students to join the Action and the research on reversible computation

O3. Cooperate with industrial partners, including developers of reversible debuggers, software testing tool producers, robotics companies, nano-electronics and telecommunication companies, on evaluating and adopting solutions delivered during this Action. This will be delivered by a cooperative work on the Action's case studies and through the ongoing work on nationally funded research projects.

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

The main objective of the Action is to coordinate, stimulate and cross-fertilise reversibility research in Europe, to overcome its fragmentation and relative isolation within small groups of researchers, and to avoid duplication of effort and to maintain and improve Europe's position in the field. Additionally, the Action aims to support knowledge transfer and collaboration links between the existing groups and new groups, and between computing community and other disciplines, and with industry. In order to achieve these objectives the following networking activities will be set up:

1. Create and maintain a website for the Action, equip it with collaboration support tools, and provide a link to Reversible Computation website (http://www.reversible-computation.org)

2. Involve at least sixty academic and research institutions in the Action. Currently, over forty institutions in sixteen countries expressed interest in this Action

3. Cooperation with industrial partners, including fact-finding visits, feedback on research, demonstrations of prototype circuits, systems and software

4. Proposals on topics in reversible computation will be submitted to national and European support agencies

5. Management Committee meetings will take place annually

6. Annual Action Workshop (AW) (associated with Reversible Computation Conference) will serve as meeting for all Action participants, with invited speakers from the related scientific fields and from industry

7. Training schools for PhD students and early-career researchers

8. There will be around 100 Short-Term Scientific Missions

9. There will be at least 160 publications in international journals, conference and workshop proceedings resulting from the research and collaboration within the Action

10. There will be a final report presenting the achievements of the Action

Working Groups will meet annually during the yearly AW, providing ample opportunity for discussions, technical presentations and demonstration of software and circuits. Each WG will compile a state-of-the-art report in the first year, and follow it with yearly update reports.

C.4 Potential impact of the Action

One of the main scientific impacts of the Action will be a better understanding of reversibility, and a formulation of unified set of main challenges, problems and solutions. Novel styles of modelling, simulation, testing and debugging will lead to more effective software tools. Reversibility-inspired theories and formal methods will enable software industry to deliver safer and more reliable distributed software and systems. They will also assist scientists from other disciplines, for example biochemistry, mathematics and material science (superconductors), thus enhancing interdisciplinary cooperation and leading to new applications.

The research in new reversible and quantum logics will lead to reversible circuits and prototype reversible hardware which, in turn, will lead ultimately to the development of low-power devices and green ICT. In order to be ready to program and operate such revolutionary computers, reversible algorithms, programming languages and software architectures will be developed during the Action.

Expected impacts from a societal and economic point of view include safer and more reliable software and systems, more robust mechanisms for dealing with faults and failures, leading to higher trust by society in systems and software, lower cost of operation and to improved quality of life. The prospect of significantly improved energy efficiency of computing is crucial, and the programme of the development of a new technology and a new way of computing will also boost overall innovation.

Increased synergy, cross-pollination of research and capacity building will maximise the benefits of national and transnational projects on reversibility. This will in turn consolidate the internationally leading position of the European research in this field.

C.5 Target groups/end users

The main stakeholders and end users who will benefit from the results of the Action are the scientific community, the software industry, the telecommunication industry and nano-electronics companies. Representatives from each of these stakeholder categories have been consulted and have contributed to the preparation of the proposal for this COST Action. Ultimately, society as a whole will exploit the expected benefits of safer and more reliable software and systems as well as of low-energy computing.

D. SCIENTIFIC PROGRAMME

D.1 Scientific focus

The Action will focus on the following research topics and challenges:

- Explore alternative forms of reversibility for finite-state computing models and investigate their precise impact on their computational and descriptional complexity

- Develop new general purpose reversible programming languages in order to better understand how reversibility impacts programming in general, and to offer abstractions not available in the more fine-grained finite-state computing models

- Develop novel compiler technology for garbage-free translations of reversible programs onto reversible machines, hence to provide a springboard for further research on reversible algorithms

- Develop new reversible process calculi that support both causal-consistent and more general forms of reversibility, with aim to evaluate them in the modelling of, for example, transactional systems, biochemical reactions and movement in robotics

- Investigate suitable notions of behavioural equivalences and logics for reversible computation, in order to facilitate model checking of properties of distributed software systems and quantum systems

- Identify the interface between physical circuit design and logical circuit design

- Develop a general model of reversible circuits that can be used, on the one hand, to implement the circuits and, on the other hand, to serve as a target of logical circuit design

- Develop methods for automatic synthesis and optimisation of reversible logic circuits

D.2 Scientific work plan methods and means

WG 1: Foundations

There exist several notions of reversibility for finite-state computing models. It is desirable to investigate their precise impact on the computational and descriptional complexity of the devices. Relations between the approaches and a clarification and classification of the different notions are of particular interest, and will be studied in depth.

Furthermore, in the literature one can find several types of automata that are equipped with additional resources or structural properties. Examples are asynchronicity for cellular automata and storage media for finite-state machines. Research on possible relations between these properties and resources and reversibility is only at its beginning and will be intensified. Since currently only little is known about these questions there is a wide field of open and untouched problems worth studying. Examples are problems concerning the decidability and universality of asynchronous cellular automata under some restrictions for cell updates, or the size trade-offs between minimal devices and minimal reversible devices.

In the domain of reversible programming languages, the aim is to move reversible programming out of the prototype language realm. General purpose languages (both imperative and functional) will be developed, and will have high-level abstractions such as dynamic data types, polymorphic types and higher-order functions. Such features will likely require incorporating sophisticated extant language techniques such as reflection as well as new techniques yet to be discovered. This will require significant theoretical developments, such as a general theory for input/output in reversible programming.

A further aim is to demonstrate the usefulness of these languages by building a basic library of reversible algorithms for standard functionalities, such as searching and sorting, and using this as a jumping-off point for reversible algorithmics. Also, a novel compiler technology for translating reversible programming languages to reversible machine code will follow. The main challenge here is that the translation is clean and efficient. This means that not only the translation should preserve the semantics of the source program but there should be reasonable guarantees about the intermediate garbage production.

Understanding the role that reversibility plays in natural systems, for example in cell biology, will help in the development of realistic formal models for concurrent and distributed systems. Further work is needed on different forms of reversibility, mainly causal-consistent versus out-of-causal-order reversibility, and how to best model systems that are only partially reversible. The mentioned models will be based mainly on process calculi, such CCS or the pi-calculus. There is, however, a need for more abstract formalisms such as reversible event structures and modal logics. Recent results show clearly that adding the reversibility feature to these models increases considerably their expressiveness. Hence, there is an expectation that further research on, for example, modal logics for reversibility can result in new formulations of useful properties of systems (safety, precedence and exception), which can lead to more efficient model checking.

It is still an open problem what are the most suitable behavioural semantics for reversibility, both for labelled transition systems and reduction-based systems. Several reversible labelled bisimulation equivalences have been studied recently but there have almost no work on reduction equivalences: this will be addressed during this Action. Testing semantics has proved very successful in industrial testing applications and the Action intends to generalise it to the reversible setting, thus giving sound foundations to commercial reversible testing and debugging software tools.

In quantum computation, although the standard quantum protocols, including cryptographic protocols such as key distribution, have been mathematically proven to be correct and secure, experience from classical computer science shows that a mathematical proof about an idealised protocol is different from verification of an implemented system. The aim of research in quantum process calculi is to develop techniques for analysing descriptions of protocols that are close to the level of real program code. This makes it possible to discover and eliminate bugs lurking in the detailed programming of a protocol, for example at the interface between classical computation and quantum communication. An ultimate goal of this line of research would be software that can analyse a protocol description, expressed in a language analogous to a programming language, and automatically report whether or not the protocol satisfies a given specification. Moreover, there are graphical languages such as, for example, the zx calculus for reasoning about quantum processes. The zx calculus is sufficiently powerful to replace the Hilbert space formalism. The aim is to study the fragment of this calculus that deals with reversible computation, aim for a full characterisation and work on completeness with respect to a variety of models. Then the resulting calculus will be implemented in a software tool called Quantomatic to assist in reasoning about reversible circuits.

WG 2: Software and Systems

Reversibility has the potential of providing linguistic abstractions, languages and tools to develop safer and more reliable applications. Also, it will simplify the development of applications where reversible behaviour occurs. While interesting results have been obtained in this area, different steps are still needed to make this possible on a more general basis.

First, the theory of reversibility needs to be extended to cope with many aspects of mainstream languages. This includes specific programming constructs such as dynamic data structures and error handling primitives such as try-catch. Also, modularity aspects such as classes and objects, or modules, or components should be considered to deal with real large-scale programs. A main shortcoming of current approaches to reversibility is that they normally do not consider distribution, but only concurrency: this limitation should be overcome. One should also understand how to integrate reversibility into current type systems. Also, behavioural type systems could be developed to control reversible behaviours.

Second, how to exploit reversibility to define actual reliability constructs and frameworks and how to program real applications should be understood. Indeed, different preliminary results show connections between reversibility on one side and some forms of transactions and checkpoints on the other side. These works should be extended to cover a wider range of existing techniques. For instance, software transactional memories are intuitively based on some form of undo, but the relationships with reversibility are not clear. The aim of these studies is to have a global theory based on reversibility and covering all these reliability mechanisms, to be able to compose them soundly (which is not the case currently), to improve them, and to develop completely new schemas.

Reversible computation is also at the core of newly-proposed programming paradigms for developing control systems and robots with a high level of autonomy and adaptation. For control systems, current proposals in this direction largely lack solid semantic foundations, which are a prerequisite for reasoning about the resulting systems. One should try to apply the existing theories of reversible computation to provide formal semantics for those programming paradigms, thus enabling the development of methods for reasoning about the correctness of the control systems designed using them. For robotics, one should understand how to make a full language for robotics reversible, and how to deal with irreversible side effects on the environment.

Finally, one should put reversibility to work to develop languages and tools for the development of reliable applications beyond the current state of the art. This requires a considerable engineering effort, covering both algorithmic aspects, such as space and time efficiency, and integration with current technologies. Enabling quick exploitation of reversibility features in software development requires the development of libraries or tools for mainstream languages or for domain-specific languages in the relevant areas, so that they can be integrated in the existing software development process. Libraries could provide primitives for exploiting reversibility-based patterns. In tools, reversibility can be used, for example for debuggers, event-based simulators and software versioning systems. Which are the good primitives allowing to use reversibility in these settings is largely to be discovered. Also, the study of causal-consistent reversibility will deepen our understanding of causality, and this can be exploited in code analysis and optimization tools.

WG 3: Reversible Circuit Design

Although reversible logic has been seen as a mainly academic issue for a long time, recent accomplishments have also triggered the interest of industry. Landauer’s Principle, and the motivation for low-power circuit design, has been validated experimentally in 2012. The first physical realisations of reversible circuits based on CMOS, adiabatic design principles, the superconductor flux logic family, and photonic technologies have been developed recently, and are attracting widespread interest from the circuits and hardware industry. As a result, firstly industrial SMEs but also big players in the semi-conductor devices industry have shown an interest in these technologies.

The aim of this Working Group is, firstly, to bridge the gap that exists today between research in realisation of reversible circuits and the research on the design of reversible computational structures. The Action will do this through Action Workshops and open seminars where the different research fields can interact. Secondly, the Action will seek to make industry more aware of the possibilities that the recent developments in the field provide. The Action will first work with the existing industrial partners with a view to growing a base of potential partners interested in reversible logic circuit technologies. The aims are described in more detail below:

Technology and knowledge exchange: Exploiting the promising applications of reversible logic, as stated above, requires joint effort from several different disciplines. In particular, logic designers and computer scientist on the one hand and physicists and electrical engineers on the other hand need to come together in order to agree and define proper interfaces between the respective domains. More precisely, models such as gate libraries, formal and discrete descriptions of constraints, and cost metrics are needed which (a) abstract the respective physical realisations for the purpose of logic design and (b) are precise enough so that the corresponding logic designs can be used as a blueprint for physical implementations. As an example, a very promising photonics technology, developed by a photonics technology research institute, based on a reversible logic has not found application in circuit logic design yet due to a lack of formal models.

Industrial awareness activities: The WG participants will employ some of the resources provided by this Action to intensify existing industrial cooperation with a view to bringing the developments in logic circuit design aimed at by this Action into the application domain. Some participants of this Action and the proposers of the scientific programme of this WG are involved in nationally funded research projects that have been conducted in a close cooperation with industry. All of these projects focus on conventional circuit design. However, this Action will allow this WG to inform and interest the industrial partners in novel technologies based on reversible logics and in the possible applications presented in Section B.2. This will be achieved by participation in relevant trade and computer technology exhibitions (see also Section C.3.).

WG 4: Case Studies

The theories, techniques and support software tools delivered by Working Groups 1 to 3 will be validated and their suitability for practical applications will be assessed in suitable case studies. The initial selection of appropriately challenging case studies was achieved during the preparation of the Action proposal, often in consultation with industrial partners. Once the Action starts, one of the main activities in the first year will be to discuss the proposed case studies, consider further case studies suggested by newly joined industrial partners, and agree a realistic timetable. The main case studies currently planned are as follows:

1. Software applications can be made safer and more reliable by employing programming language abstractions, languages and software tools based on and benefiting from the reversibility feature. Software applications will be considered in this case study that involve transactions or checkpoints such as, for example, software transactional memories and long-running transactions with compensation.

2. Reversible debugging software tools are very successful for debugging sequential programs, where the reversibility boils down essentially to backtracking. When faced with multi-threaded programs, it is less clear what is the logically and operationally correct form of reversibility. The Action expects that causal reversing is appropriate but the situation is made more complicated by various optimisation tasks performed during execution of multi-threaded programs, which effectively change the program order. Hence, the understanding of reversibility of multi-threaded programs for the purpose of debugging must be reached in the setting of so-called weak memory models such as, for example, the weak memory model of Java. Also, a big issue for reversible debugger developers is performance of their software tools, particularly of multi-threaded recording. They are also very interested in using reversible replay debugging to provide new ways to explore a program's behaviour.

3. Programming of robotics is a natural application area with a number of potential applications such as industrial robotics, mobile robotics and self-reconfigurable robotics. Automatically reversing a sequence of operations brings a robot back to its initial shape, as has been experimentally demonstrated using the DynaRole reversible language. DynaRole, however, only allows simple sequences of operations to be

reversed, which is suitable for reversing self-reconfiguration sequences but lacks the generality needed to implement more complex behaviours. This case study will generalise DynaRole to support a wider range of modular robot control scenarios, while retaining the possibility of reversing distributed sequences. Here, reversibility will be investigated as a practical feature, aimed at reducing the programming task, and allowing error recovery by backing out of error states using reverse execution. The aim is to employ the extended DynaRole language in demonstrating the use of reversibility in large-scale simulated scenarios as well as small-scale physical experiments.

4. Modelling of biochemical reactions, where undoing of bonds between participating molecules plays a crucial role in driving the reaction and reaching its end result, will test the achievements of this Action in developing useful process calculi for reversibility. A real-world example of a biological system suggested by biochemistry experts is the isomerisation of glucose to fructose. The reaction is a step in the overall glycolysis pathway which transforms glucose, also known as grape sugar (a carbohydrate which is consumed with food), into pyruvate forming the high-energy compound ATP (adenosine triphosphate) and NADH (reduced nicotinamide adenine dinucleotide), which can then be used in various other reactions. In addition to the modelling of creation and dissolution of chemical bonds, spatial organisation of molecules needs to be represented, as well as their movement.

5. Demonstration versions and prototypes for reversible circuits, or one of their applications such as quantum circuits, are costly to provide and, hence, will not be considered within this Action. Instead, the Action shall demonstrate the applicability of the reversible computing paradigm in several application domains by means of technology simulators. In particular, simulation of quantum systems as well as photonic systems will be considered for this purpose. Using such simulators, the technical background and models as well as the suitability of the corresponding design methods and technology mapping schemes, as developed in WG3, can be demonstrated. If new technological constraints emerge, they can directly be incorporated in the simulator and, hence, immediately validated. This allows for a direct incorporation of new results obtained in discussions between the different disciplines. At the same time, these simulators can be used to demonstrate the applicability and the promise of these technologies to industrial partners.

The case studies will provide feedback on the effectiveness of the reversibility-inspired theories, techniques, and solutions, and will point to new applications and research questions.

E. ORGANISATION

E.1 Coordination and organisation

The organisation of the Action and coordination of its activities will follow successful experience of previous actions, and will be in accordance with Rules of Procedure for Management Committee. The Action will be coordinated by the Management Committee (MC), which will be presided by Action Chair, assisted by Vice-Chair and Secretariat. MC will meet once a year jointly with the annual Action Workshop (also see below).

The scientific activities of the Actions will be carried out in four Working Groups, as given in Section D.2, led by Working Group Chairs and Co-Chairs, who will be appointed by the MC. They will participate in the MC meetings to report on progress of their WGs. Each WG will meet once a year and, at other times, will use a forum on the Action's website (see below), email and Skype to discuss work and the affairs of their WGs.

To promote participation and training of young researchers, the Action will support generously Short-Term Scientific Missions (STSMs). These are mainly intended for PhD students and early-career researchers. However, since one the aims of the Action is to educate participating researchers in the expertise of other groups and to eradicate fragmentation of research, there will be a number of STSMs for more experienced researchers. There will be a small STSM committee appointed by the MC to approve STSMs.

There will be annual Action Workshop which will be collocated with the RC event if it takes place in Europe. AW will be open to all Action participants and any interested people, and will feature invited speakers from scientific and industrial institutions.

The Action will organise two training schools, likely in Years 2 and 4, aimed mainly at young researchers but also suitable for scientists from related fields and industrial participants.

There will also be the Action's website, containing a general discussion forum and forum areas for the MC and for each of the WGs. The Website manager, appointed by the MC, will be responsible for the website. The website will also support dissemination of the results of the Action, see Section

H.

Milestones of the Action are listed below. They will assist the MC in checking the progress towards the achievement of the Action's aims and objectives.

- M1, end of Year 1: State-of-the-art reports by WG 1 to 3; report from WG 4 on planned case studies with feedback from industrial partners; First Annual Reports from WGs and First Annual Report of the Action; Proceedings of the first AW; the Action website operational

- M2, end of Year 2: Second Annual Reports from WGs and Second Annual Report of the Action; Proceedings of the second AW; Publication of the training materials for the first training school

- M3, end of Year 3: Third Annual Reports from WGs and Third Annual Report of the Action; Proceedings of the third AW

- M4, end of Year 4: Final Annual Reports from WGs and Final Annual Report of the Action; Proceedings of the fourth AW; Publication of the training materials for the second training school

E.2 Working Groups

There are four Working Groups: WG 1 Foundations, WG 2 Software and Systems, WG 3 Reversible Circuit Design and WG 4 Case Studies (also see Section D.2). The membership of each WG will be decided mainly by the interests and expertise of the Action participants. It is expected that, due to the character of WG 4, most participants of the Action will belong to WG 4. Since the themes of the first three WGs are traditionally fairly disjoint, and in order to provide efficient transfer of knowledge, agreement of the basic principles of reversibility and to avoid duplication of effort, the MC will ensure that many Action participants, including Chairs and Co-Chairs of WGs, belong to more than one WG. Moreover, cooperation between WGs will be helped by a collaborative work on the case studies of WG 4 and by regular meetings between the Chairs and Co-Chairs of WGs and with the MC.

E.3 Liaison and interaction with other research programmes

The majority of this Action's research is disjoint with activities of other European projects although, as is described in Section B.4, some of our specific aims are shared with the projects HATS, ENVISAGE and BETTY. The currently running projects ENVISAGE and BETTY share a small number of participants with this Action. When a need arises other members of ENVISAGE and BETTY will be invited to participate in this Action's WG meetings. They will also be welcome at our annual workshops and RC conference.

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

Gender balance is one of the facets of computer science that must be improved. Female members of student body and staff of computer science departments of European universities and research institutions make up only between 15% and 20% of all members. This Action prides itself in addressing this imbalance from the start: several leading female researchers have been involved in the preparation of the Action, and a number of other female scientists, including PhD students and early-career researchers, have agreed to join the WGs of the Action. The list of experts in Part II contains 18 female members out of 63, meaning that over 28% of the members of this Action are women. There are also other female PhD students within the groups that make up this Action, and the MC and WGs will work hard to involve all female participants fully in the activities of the Action.

Reversible computation is a fairly new and fast-growing discipline in ICT. It spans the whole range of types of research, from theory and programming to circuits and hardware. Its novelty and promise have attracted many young researchers, and more and more PhD students are choosing to research reversibility. Not surprisingly, most of the networking activities of the Action are geared up to support them (see Section E.1). Over 36% of the proposed participants in this Action are either PhD students or early-career researchers. Many of them, through their ground-breaking research, are already recognised experts in the field and will become members of the MC or will lead some WGs, and will also present courses at training schools.

F. TIMETABLE

The duration of the Action will be four years. The first gathering of the Action's participants will be at a kick-off meeting where the MC will be elected. The MC then will discuss and appoint the Chairs and Co-Chairs of WGs, the Action's website manager, and STSM committee. The MC and WGs will meet once a year, typically at the venue of the Action's Workshop. These meetings will be over two days so that there is ample time for the MC meeting, technical meetings of WGs and participation in sessions and discussions of the workshop. Three months after the kick-off meeting, the Action website will be launched.

In Year 1, the Action will work on the state-of-the-art review of reversible Computation research, and on selecting suitable case studies. The report resulting from this review and selection of case studies will become a part of the First Annual Report of the Action at the end of Year 1. This report will also contain a detailed updated plan of work for each WG and for dissemination of the results of the Action. The Action's activities will then be reported in Annual Reports in Years 2 and 3, and in the Final Report at the end of the Action. There will be an open Action Workshop in Years 1, 2, 3 and 4, and there will be two training schools in Years 2 and 4.

	0	1	2	3	4
Kick-off meeting	*				
Action website launched	*				
Case studies finalised		*			
MC meeting and WG meeting		*	*	*	*
Action Workshop		*	*	*	*
Training school			*		*

G. ECONOMIC DIMENSION

The following COST countries have actively participated in the preparation of the Action or otherwise indicated their interest: BE, CY, DE, DK, FI, FR, IE, IS, IT, NL, PL, PT, RO, SE, SI, UK. On the basis of national estimates, the economic dimension of the activities to be carried out under the Action has been estimated at 64 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?

The target audiences for dissemination of the Action's results are

- International research community working on reversible computation, including participants of this COST Action
- Researchers in the related fields, for example material science (superconductors), electronic engineering, cell biology, mathematics and quantum physics
- Software, circuits and hardware developers within industry
- The general public with an interest in novel technology and green ICT

H.2 What?

The Action will disseminate its results through

1. The Action's website will contain reports on activities, events, research breakthroughs and publications, workshop and training school's reports and grant applications
2. A web-based communication network for Action participants, in particular for PhD students and young researchers to help them to establish collaboration links
3. Annual AWs open to all interested in reversible computation
4. Two training schools and many Short-Term Scientific Missions for Action participants
5. Publication of articles in peer-reviewed conference proceedings, for example in Reversible Computation Conference (http://www.reversible-computation.org) and in journals
6. More popular forms of publications, for example in general computer science publications and in national magazines

7. Presentation and demonstrations of software and circuits at scientific conferences and events, at topic-specific expos such as, for example, the Design, Automation and Test in Europe conference, as well as also at broader events such as, for example the CeBIT, the world's largest international computer expo.

8. Invitation to demonstrations of experimental hardware and software being developed by the Action in cooperation with industrial partners

9. Supporting PhD students and early-career researchers in seeking internships with software and technology companies who are interested applications of reversibility

10. Annual Reports by the WGs and the MC

11. The Action's Final Report that describes the outcomes and achievements of the Action

H.3 How?

Scientific results and technological solutions obtained by the Action will be disseminated via the traditional forms of dissemination of knowledge, namely through refereed scientific journals and conference proceedings.

The Action website will inform scientific community and the general public about the Action's activities and achievements, and will include pointers to publications, events and meetings. The website will also be used by Action participants to discuss ongoing research and to announce preliminary results and draft papers.

The annual workshops, with special sessions devoted to presentations by industrial partners, and Reversible Computation Conference will be the main venues for presenting work on reversible computation.

Exchange and fact-finding visits by Action participants, in particular involving PhD students and industrial partners, will contribute to the exchange of knowledge and expertise.

Demonstrations of novel circuits, hardware and software will increase the awareness and knowledge of Action's achievements among members of industry as well as among the general public

Some of the research results, together with background material, will become teaching materials for advanced university courses and for courses given at the Action's training schools, helping the education of young scientists and preparing them for revolutionary future technologies.