

DIGIT Project description

Objectives

Recent years we have seen an unprecedented acceleration in the evolution and expansion of Internet-based service provision and inter-intra-enterprise integration in all market sectors. This brings about the prospect of an ad-hoc integration of systems across organisational boundaries. Spanning national and enterprise borders, the participating entities pool resources, information and knowledge in order to achieve common objectives. Such new collaborations may last for a single transaction or evolve dynamically over many years. This expected market development sets new requirements for interoperability and trust, necessitating the on-demand creation and self-management of dynamically evolving systems. A demanding and timely research challenge is the provision of cost-effective trust and contract management solutions that enable interoperability and secure collaborations in such systems of systems.

Overall goal: Build a leading applied research group nationally as well as an internationally at SINTEF and UiO targeting interoperability and trust through the development of a framework facilitating cost effective and reusable interoperable digital trust and contract management solutions as well as analysis methodology for systems of systems.

Sub-goals:

- (1) Development of a framework as specified above including (A) a superlanguage for modelling systems of systems, with particular focus on digital interoperability and trust, (B) conceptual models and ontologies, (C) specialised metamodels and languages, (D) an open source based design environment, (E) support for various execution environments, including executable models, and (F) a methodology.
- (2) Documentation of the framework and its methodology in the form of a handbook.
- (3) Dissemination and exploitation of results among the industrial partners through a trial-driven process.
- (4) Dissemination of results, both nationally and internationally, through publications, public workshops, standardisation, future national and international projects, and university lectures.
- (5) Assessment and evaluation of DIGIT results.
- (6) Graduation of three PhD fellows.

Frontiers of knowledge and technology

Interoperability is defined as the ability of two or more systems to exchange information and to meaningfully use the information to work towards common objectives. Such interoperability will occur only if the parties involved are willing and able to interoperate. This means that in addition to focusing on the ability to interoperate, one must also consider the mutual trust relationships between the involved parties, i.e. whether they are willing to interoperate. DIGIT will focus on mechanisms necessary for one party to entrust information and control to another party's IT system. Trust between two IT systems is defined as the belief of one system that the other system will do what it has promised to do in the presence of risk. This definition refines the general notion of trust to be an understanding that is negotiated, established and enforced automatically between IT systems.

Interoperability

A number of Interoperability frameworks and reference models have evolved during the last 10 years in various standardisation bodies and European IST projects, from the ECMA Toaster model, the IDEAS reference model, and through the more recent INTEROP and ATHENA Interoperability frameworks.

The Interoperability Reference Model shown below is derived from the ATHENA Interoperability framework, and illustrates interoperability issues between two systems, either in the same (intra) enterprise or in different (inter) enterprises. The context for each system can be described through Enterprise models (from the Enterprise Architecture field), and each system can be described from different viewpoints (process, service, information, non-functional aspects) on different abstraction levels, i.e. from platform independent to platform/technology specific. The vertical model-driven interoperability box illustrates that a model-based approach can be taken in all these areas, and similarly the vertical ontologies and semantic interoperability box illustrate that models in each of these areas can be semantically enriched through proper use of ontologies.

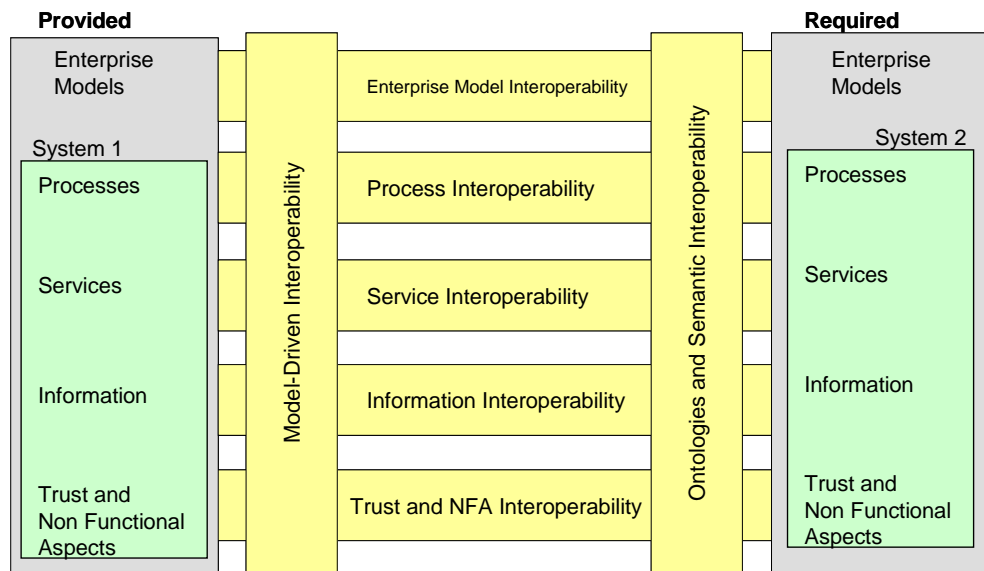


Figure 1 Interoperability Reference Model

A cluster of EU projects has developed a unified roadmap for research challenges concerning Interoperability. Based on work during the last few years and the current status, this roadmap identifies gaps between status and issues as future research challenges for interoperability. The involved projects are ATHENA, INTEROP, CrossWork, ECOLEAD, NO_REST and TRUSTCOM. The technical issues are structured into the categories of the figure above. The main conclusion for further challenges is that although we currently have isolated technology-specific solutions for the areas above, they still need to be abstracted to a model-level and language-level in order to support the flexibility required for dynamic adaptation in evolving environments.

The current state of the art and the current practice are isolated model-based mappings from research projects or technology-based mapping solutions by EAI (Enterprise Application Integration) vendors, for the moment mostly concerning the ESB (Enterprise Service Bus) concept. The DIGIT vision, beyond the current state of the art, is to use a unified model-driven

interoperability approach based on a common superlanguage and appropriate semantic mappings and mediation support.

The notion of trust

Studies of trust distinguish between the trustor, i.e. the agent that trusts another agent, and the trustee; the agent being trusted. Trust is a property of the trustor, whereas credibility and trustworthiness are properties of the trustee. Trust can be seen as a binary relation, from the trustor to the trustee. Mayer et al.¹ define trust as the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other party will perform a particular action which is important to the trustor, irrespective of the ability to monitor or control that other party. Properties of the trustee, such as credibility and trustworthiness are considered important factors influencing an agent's trust in another party. In general, there is no accepted definition of the term trust (as for example highlighted at the 2006 Conference on Trust Management²); in fact, there seems to be many different kinds of trust and these various kinds need to be properly defined and related.

Trust and security

The notion of trust is interwoven with notions like security, data protection and privacy. These notions are all of major importance to the ICT field in general, and to domains like e-commerce, e-government, e-health and in particular virtual organizations. To understand and deal with one of these notions we must also understand and be prepared to deal with the others. For example, accountability is an important aspect of integrity, and therefore an aspect of security. However, measures to achieve accountability (and in particular, the special case of non-repudiation) may easily conflict with rules and regulations for data protection and privacy. Furthermore, the implementation of security measures may considerably reduce user-friendliness and thereby affect the general notion of user trust. It is generally accepted that trust is a more general issue than security in particular and than dependability in general. For example, Jones et al.³ argue that although businesses and consumers may consider underlying systems to be completely dependable in the traditional sense, they may not trust these systems with their business or personal interests unless there is a suitable legal framework they can fall back on, should problems arise. An analysis of trust will therefore encompass a number of issues like legal, sociological and psychological aspects that are not explicitly embedded in the concepts of dependability and security.

Trust and interoperability

The emergence of web services technology as a means for establishing ICT interoperability across enterprises has introduced a number of open standards for security interoperability for web service transactions across different administrative and trust domains. Standards include SAML for expressing and distributing security assertions and WS-Security/WS-SecureConversation for enabling secure web service transactions, which are both being standardised within OASIS and

¹ Mayer R.C., Davis, J.H., Schoorman, F.D. An integrative model of organizational trust. *Academy of management review*, 20 (3): 709-734, 1995

² Proc. 4th International Conference on Trust Management, LNCS 3986, 2006.

³ Jones, A. J. I. *The open agent society*, chapter 3; a logical framework. John Wiley & Sons, Chichester, UK, 2004

which have been incorporated in commercial web services middleware. More recently, a consortium of companies led by IBM and Microsoft proposed a further collection of web services interoperability protocols that includes WS-Trust, WS-Federation and focus on establishing federations of different trust domains as well as brokering trust between these domains. Although some preliminary experimentation with such protocols for Single Sign On has been initiated, their use for brokering trust across different DTFs⁴ has not been addressed yet. In parallel, a set of interoperability protocols including WS-Coordination and WS-Transaction has been proposed by the same consortia for automating transactions distributed across different administrative domains. On the other hand, the integration of web services security and transaction protocols for securing distributed transactions across different DTFs has not, with the exception of a few preliminary studies, been addressed as yet.

Trust services

With the growing number of Certificate Authorities (CA) across the globe, Trust Services vary more than ever. Trust Service markets take different forms depending on the shaping effects of the jurisdictions that host them. Some Governments develop accreditation schemes, such as UK's tScheme, with strong shaping effects into how Trust Services are deployed; others follow a more "soft" approach leaving markets dynamics and standards as control mechanisms. These different jurisdictional approaches foster differences in liabilities, Certificate Practice Statements (CPS) and the final products/services. There is also a large variety of technologies and processes deployed by Trust Service Providers, ranging from types of technologies, registration processes, revocation model, etc. Attempts to foster homogeneity in Trust Services through enforcing technical, operational and legal standards have not solved the problem.

Models and execution

The proof of the pudding is in the eating. The semantics of systems is defined in terms of how the systems execute, not (only) how they are specified. Traditionally, this has been evident for systems specified by programming languages such as Java or C++, but it has been less obvious for systems specified by modelling languages such as the UML. Without model execution, models will not be the central artefact in systems development, but will rather remain an adjunct. Model execution enables the understanding of the system's architecture and its implications early in the development cycle, it enables the early simulation and testing of applications, and it helps to bridge the IT-to-business communication gap.

Current behavioural modelling relies mostly on state diagrams for the description of behaviour in a generic way (e.g. Mellor and Balcer⁵, Rose RT⁶, Rhapsody⁷, AGEDIS⁸). Sequence diagrams are widely used to describe specific behaviour patterns but since they only give a partial description of the system, more information must be added to achieve a complete execution.

⁴ DTF = domestic trust framework

⁵ Mellor, S.J. and M. Balcer, Executable UML: A Foundation for Model-Driven Architecture. 2002: Addison Wesley. 416.

⁶ IBM Rational Rose RT was a product of Rational: www.ibm.com/rational

⁷ Rhapsody is a product of i-Logix.www.ilogix.com

⁸ AGEDIS is an IST-project: www.agedis.de

Harel's Play-In Play-out technology⁹ attempts to simulate execution of the Harel variant of sequence diagrams (Live Sequence Charts).

UML2 activity diagrams now support control and data flow with combined token flow semantics inspired by Petri nets. This means that activity diagrams can also be used as a means of executable definition of behaviour.

Execution platforms normally support only one approach such as Windows, Eclipse, CORBA, EJB or similar. Already these platforms run on top of each other or beside each other, but there is no unified conceptual execution platform.

Composition

The composition of systems into larger systems ranges from static composition (supported by compiler-like tools for composing models/programs) to dynamic composition. Most approaches are based on an underlying understanding of system architectures in terms of runtime components with ports or interfaces and with connectors between these.

The means of specifying composition (or architectures) ranges from frameworks in programming languages, via extensions of single programming languages (like ArchJava¹⁰) to separate languages for specifying the composition from the languages of the different (sub)systems. The last category includes architecture description languages (ADLs¹¹), general-purpose modelling languages like UML¹² (with 2.0 especially supporting ports and connectors) and more specific languages, for specifying Web services.

Model composition and related abstraction and refinement techniques are now emerging as key mechanisms for managing scalability in complex systems modelling. However, one technique has not yet been developed, namely that of assisting the development of large-scale models through independent development, compilation and composition of partial models. Such a technique would be similar to Parnas' influential module concept.

Although there is no new module concept emerging, the notion of megaprogramming has been introduced, in order to program by means of megamodules. The idea is that these would typically belong to different organizations, made in different languages and with different type systems. This was later taken up by the CHAIMS¹³ project at Stanford, where a language for megaprogramming was developed. In 2000, the corresponding notion of megamodelling was introduced.

Mechanisms have been devised for combining different types of system views, reflecting information relevant and interesting to different stakeholders (e.g., business analysts, hardware engineers, certifying agencies). These views are possibly represented in different languages or at different levels of detail. When composing such models or producing a unified model, redundant overlaps must be reconciled and consistency must be checked. A similar approach is represented by Aspect-Oriented Programming, the idea being to tackle the complexity of a system by partitioning it into aspects (collected in modules) that are woven together to comprise the full system (e.g. AspectJ¹⁴, CME¹⁵). Although most efforts in this area address the programming

⁹ Harel, D. and R. Marely, Specifying and Executing Behavioral Requirements: The Play-In/Play-Out Approach. *Software and System Modeling (SoSyM)*, 2003. 2(2): p. 82 - 107.

¹⁰ www.archjava.org

¹¹ www.sei.cmu.edu/architecture/adl.html

¹² www.uml.org

¹³ www-db.stanford.edu/CHAIMS

¹⁴ eclipse.org/aspectj

world, some also address models (e.g. Theme/UML). Both systems' views and aspect orientation has a module concept that crosscuts the main components of a system.

The state of the art in static model composition is predominantly theoretical; only limited prototype tool support exists, such as the Atlas Model Weaver¹⁶ and the GGT toolset¹⁷ from LIP6. These tools typically deal with syntax-based unification of models that conform to the same metamodel without semantic integration. Furthermore, it is questionable whether the current state of the art supports the scale and level of complexity necessary for engineering complex systems.

Most development material is currently stored in tool-specific file formats and the version management is done in systems like Subversion, CVS or ClearCase. Configurations first need to be established across tool boundaries. Then, different versions of models and other artefacts need to be merged. An approach like MOF and MOF versioning may help here, but further support is required.

There is hardly any established way of doing dynamic composition of systems. The need for doing this has been recognized, but it is still subject to ad-hoc solutions. Even the composition of single systems still relies on systems consisting of objects – the idea of component is still only a model/program structuring concept and not a runtime concept. Some research languages, like Scala¹⁸, experiment with (large) objects playing the role of runtime components.

Research tasks

Current practice for system modelling supports neither (A) the development nor (B) the analysis of complex systems of systems to a sufficient degree, especially when trust is essential. The entire history of software engineering and analysis is one of rising levels of abstraction, since abstraction is the primary way for humans to deal with complexity. This tendency is reflected in the maturation of our programming languages, analysis methods, platforms, processes, tools, and patterns. DIGIT will continue this tradition by searching for useful abstractions for (A) developing and (B) analysing systems of systems with trust.

Technically we may formulate the research challenge as the development of a framework of improved methods, tools and languages with the overall objective to facilitate cost effective and reusable interoperable digital trust and contract management solutions for systems of systems. To this end, we will search for a *superlanguage* for modelling systems of systems, with particular focus on digital interoperability and trust. While an ordinary language has language primitives representing system elements (an attribute, a loop, an interface), we believe the superlanguage should have complete system executions as its primitives. The superlanguage should have a semantics that is appropriate for modelling, executing and analyzing systems of systems. The superlanguage platform will rely on the supporting platforms of the languages of the primitive systems. The superlanguage will extend the current system development technologies to a new generation of technologies capable of supporting interoperability and trust.

The achievement of such a framework and its supporting superlanguage is extremely challenging. The research challenges include:

¹⁵ www.research.ibm.com/cme

¹⁶ www.sciences.univ-nantes.fr/lina/atl/AMMAROOT/AMW/

¹⁷ Bouzitouna, Gervais, Blanc: Models Reuse in MDA: In Proc. SERP 2005

¹⁸ <http://scala.epfl.ch/index.html>

- Defining flexible concrete syntax based on a common metamodel supported by precise and modular semantics as well as principles for composition
- Modelling and analysing mutual dependency between systems
- Understanding risk and trust management at the level of systems of systems as well as its impact on the superlanguage
- Understanding interoperability and trust preserving transformation from platform independent descriptions to platform dependent descriptions, as well as means of defining transformations of the exchanged data
- Developing methods and tools for analyzing digital interoperability and trust, and for documenting results
- Enable reuse and instantiations of generic solutions for interoperability and trust.

It is highly unrealistic to believe that DIGIT will solve all these challenges, but the DIGIT framework will be developed with the aim to get as close as possible to solving them. Which ones of these challenges will receive the most attention will depend on the views and opinions of the industrial steering committee of the project (see Section 5.1). Their opinions at the various stages of the project will depend on the results and experiences from the industrial field-trials (see the iterative research process driven by industrial trials described in Section 7). However, the framework will integrate results and provide a common reference. The interoperability framework will consist of coherent methods and tools that are useful for analysing and improving the interoperability characteristic of a system; issues related to trust and security will be given special attention.

The framework will establish the criteria for successful, model-based interoperability and trust by identifying criteria, guidelines and patterns that will leverage the ability to analyse and be ready for interoperability. This includes:

- Gathering requirements and state-of-the-art related to methods and tools for supporting model-driven interoperability and trust
- Defining the criteria, guidelines and patterns to support interoperability and trust requirements
- Synchronizing and gathering the work from the PhD-fellows
- Defining suites of techniques to support interoperability and trust requirements
- Establish an integrated interoperability and trust framework

In order to be successful the PhD-fellows will require stability with respect to their PhD-topics. Thus, their work should not be influenced by the steering committee to the same extent as the more applied research. The topics for the three PhDs are described below. These will be described in further detail depending on the interests and talents of the respective PhD-fellows. All three PhD-fellows will work 75% on their PhDs and 25% on industrial activities within the project (over 4 years) to make sure that they get a sufficient understanding of the practical needs that industry sector has.

PhD-Topic I: Interoperability and trust

Abstract models have undetermined properties that may be determined in through refinement. We say that two abstract models are interoperable if such refinements of both models exist such that

implementations of those refined models interoperate. DIGIT will investigate how one can determine whether two abstract models are interoperable, as well as how one can preserve this property through the refinement process, while at the same time maintaining relevant aspects of security and trust. In model-driven development, this corresponds to choosing an appropriate model transformation to preserve the interoperability characteristic, but in addition we have the requirement that these transformations must be security and trust preserving.

The PhD-fellow will develop methods for analysing interoperability of model refinements, and tools that can support the automation of appropriate mechanisms to provide interoperability. Automating cross-component bridges and mediators that enable communication and bridge semantic gaps is one approach of automation, combined with interactive mapping editors for human resolutions. Important research issues include:

- Defining the foundation for model-driven interoperability, such as security and trust preserving refinement techniques, model transformation, and mechanisms for traceability between model abstractions.
- Establishing methods and techniques for defining and analysing refinements and interoperability characteristics of refinements.

PhD-Topic II: Models and executions

DIGIT will perform basic research in the definition of a new group of languages called superlanguages and provide clear distinctions between a superlanguage on the one hand and other languages such as metalanguages and megamodelling languages as well as traditional modelling and programming languages on the other hand. DIGIT will in particular invent a specific superlanguage for interoperability with trust. This superlanguage must be provided with a flexible, concrete syntax based on a common metamodel supported by precise and modular semantics along with principles for composition.

A major challenge is to make the superlanguage adaptable such that the language support tools change relative to the environment in which they work. By the environment we mean the kind of bottom level systems that the superlanguage will have as its primitive objects. We foresee that the superlanguage will have both graphical and textual formats, and it may run on Java as well as other programming platforms. Major research and PhD work will address:

- Definition of a superlanguage – concrete and abstract syntax, precise semantics
- Superlanguage execution machinery – the execution platform, a platform of platforms
- Static and dynamic means of interoperability – properties of reflection
- Analysis of systems of systems – multilevel analysis tools, theory and implementation

PhD-Topic III: Composition

DIGIT will investigate how to scale up existing mechanisms for composing single systems to cater for composing systems of systems, and will focus in particular on issues related to security and trust. Mechanisms for composition come in two flavours: static composition of model/program fragments and dynamic composition of systems (i.e. model or program executions).

- Well-known mechanisms like role synthesis, aspects and even elements of architecture description languages are mechanisms for programming/modelling “in the small”¹⁹. It will be of interest to investigate whether they can be applied to programming/modelling “in the large”, maybe even for dynamic composition, and to what extent they fulfil the additional needs of security and trust.
- Mechanisms for mega modelling and programming are really designed for static composition. DIGIT will look into corresponding mechanisms for dynamic composition of systems modulo security and trust.

These issues will be studied both in the context of existing modelling and programming language (by making appropriate frameworks) and in the context of new language mechanisms (of existing languages or in a new language for handling systems of systems).

Research method

Experimental Computer Science and Engineering (ECSE) is defined as “the building of, or the experimentation with or on, nontrivial hardware or software systems”²⁰. ECSE is a synthetic discipline, studying phenomena created by humans rather than those given by nature, and there is much room for creativity and few direct physical constraints. The primary focus of ECSE is on artefacts – software and/or hardware which is the subject of study, apparatus used to conduct the study, or both. Often a significant part of the intellectual effort of the experimental research is embodied in the artefact. Artefacts in ECSE can have one of the following three roles in the research:

- Proof-of-performance. The artefact shows that a certain performance can be achieved, or that it is in some other measurable way an improvement of previous implementations. The results are usually quantitative.
- Proof-of-concept. The existence of an artefact in this role proves that a concept is possible to realise, at least in one configuration. The artefact is usually too complex to derive the behaviour of by only using logical reasoning or abstract argument.
- Proof-of-existence. An artefact in this role demonstrates a new phenomenon through its existence which is impossible or difficult to grasp only through documentation. A historical example of this is the computer mouse.

In DIGIT we will develop artefacts as proofs-of-concept. The artefacts will be innovative technologies that realise the framework for interoperability with trust as we previously presented. These new technologies will be validated against requirements from end users. To this end DIGIT will be conducted according to an iterative process where new ideas and artefact prototypes are tried out in industrial field trials with intervals of nine months. The results from these field trials will indicate the extent to which the expectations are likely to be fulfilled and will thus help direct the next iteration of research.

Created: June 29, 2007. Last updated: June 29, 2007.

¹⁹ The SWAT project already addresses novel “in the small” development techniques

²⁰ National Research Council. “Academic Careers for Experimental Computer Scientists and Engineers”. National Academy Press, 1994.