ITEA Technology Roadmap

for Software-Intensive Systems

2nd edition, May 2004

ITEA

INFORMATION TECHNOLOGY

FOR EUROPEAN ADVANCEMENT

ITEA Technology Roadmap

for Software-Intensive Systems

2nd edition, May 2004

Executive Summary

ITEA

INFORMATION TECHNOLOGY

FOR EUROPEAN ADVANCEMENT

ITEA thanks the Roadmap Core Team and all those involved for their valuable input. We feel that their dedication and hard work have significantly strengthened the foundations of software development for European industry. We also thank the organisations that supported the work financially.

Executive summary

This is the second edition of ITEA's Technology Roadmap for Software-Intensive Systems.

This key strategy document, which is intended as a companion to the White Paper defining the ITEA Programme[1], develops the shared vision of the **technological direction** for the programme itself; it is ITEA's "*tool for monitoring the technology changes in and around ITEA's participants' core areas of interest and steering the direction of the programme (...)*." [2]

Since the first edition of ITEA's Technology Roadmap on Software Intensive Systems was published in March 2001, the pace at which society has been "going digital" has continued to accelerate. Because a key factor in this evolution is now software technology – whose own development is strongly influenced by feedback from changes in society – we have decided to revisit the technology challenges and perspectives, and to update the findings of the first edition.

However, the questions we attempt to answer in this new edition are just the same as those put forward in the first:

- *"Which of the problems and software technology challenges tackled in the ITEA Programme need to be solved?"*
- *"How do we think they are likely to develop in the years to come?"* [3]

Work started one year after the publication of the first Roadmap, and took two years to complete. We followed the same approach as before, deepening and broadening the topics discussed in the first edition.[4] The precise **process** we used is described below, followed by a classification of software-intensive systems designed to make the analysis **meaningful** and a list of the human and other resources needed to make this 2nd edition of the Roadmap **successful**.

The process of elaboration that proved successful in the first edition has been used again.

This second edition is driven by demand for a new generation of applications. It therefore describes not only the technological challenges that confront the European software industry, but also the steps that need to be taken to meet the needs of industry, government and the business community, as well as those of citizens and customers.

To achieve this, we adopted the following **process**:

- We started out by developing scenarios which describe the potential evolution of applications in the various domains of software-intensive systems, and which define the technological challenges that will have to be met if these transitions are to be successful.
- Because applications from different domains share technologies ("convergence"), we merged and clustered the results of the scenarios, outlining the likely evolution of software-intensive systems in the years ahead.

1 *IRIS Book, which can be downloaded from www.itea-office.org*
2 *From the IRIS Book section 2.1.1, page 19*
3 *From Technology Roadmap on Software Intensive Systems, Edition 1, page 11, downloadable from www.itea-office.org*
4 *In the first edition of the roadmap, some issues related to applications and technologies were less discussed than others and this imbalance needed to be addressed. Therefore, the development of this version started with a complete reworking of two so-called "application domains" and a deep review of the other three. All elements of this work are in the appendices to the main report.*

To make it **meaningful** and to structure the process, we used two classifications: five **Application Domains** [covered in Appendices 2 to 6] and four **Software Technologies Clusters** [covered in Chapters 2 to 5]. Together, these cover the field of software-intensive systems as completely as possible, keeping some distance both from specific products (so as to avoid potential conflicts regarding Intellectual Property Rights [IPR]) and also from current trends (e.g. in architecture or computer languages).

Both editions of the Roadmap use the same Application Domains, Technology Clusters and Technology Categories which are used to classify ITEA projects.[5] The main Technology Clusters are arranged around four basic questions:

1. *Which end-to-end technologies are required to acquire, process and store content?*
   The **CONTENT** cluster is articulated in three technology categories [see Table of Contents], and deals with signals, data, information, documents and knowledge from capture to complete processing.

2. *Which technologies are required to transport and distribute content?*
   The **INFRASTRUCTURES & BASIC SERVICES** cluster, articulated in four technology categories, deals with transport mechanisms and protocols, as well as with the management of networks (including security).

3. *Which technologies are required to build effective user-system interfaces?*
   The **HUMAN-SYSTEM INTERACTION** cluster contains a single technology category, which deals with the interaction between human beings and the appliances and systems that support the services.

4. *Which technologies are required to engineer software-intensive systems?*
   The **ENGINEERING** cluster is articulated in three technology categories. It explores the complexity of engineering processes and deals with the creation of end-to-end services.

To make it **successful**, the Roadmap 2 process needed the full support of the ITEA constituency.

- Members of the various ITEA bodies provided experts to join the Core Team in charge of elaborating working documents; 17 high-level individuals came together in 15 two-days meetings and contributed regularly to a dedicated website. To validate and enhance the documents, the Core Team held four workshops, meeting the ITEA Steering Group and other experts from industry – over sixty people in total.

- During two 2-days workshops, high-level experts from top European universities, research centres and similar national programmes reviewed and supplemented the intermediate and final results.

- Public Authorities in the Netherlands and France provided financial support over the two-year period.

Evaluation of a range of technologies identified the main issues and challenges for software-intensive systems.

The new Roadmap presents its findings in terms of four technology clusters. These are summarised in the following sections, first with a brief presentation of the cluster, followed by a table, which briefly recaps the main challenges. A final paragraph presents the main conclusions.

[5] *With some fine-tuning of names and definitions over time. For an introduction to these see the IRIS Book, which can be downloaded from www.itea-office.org.*

CONTENT

In general, content is whatever is exchanged within the environment of the system, or between systems. It is processed, stored, managed and transformed. Content ranges from analogue signals to huge multimedia data depositories.

Content categories	Category definition	Major challenges
Content Acquisition & Processing	Technologies that are relevant to acquiring, transforming and modifying content.	Digital sensory system; capturing and managing contexts; efficient analysis of data, integration of information.
Content Representation	Technologies for representing and structuring data while at the same time making the most appropriate and efficient use of resources.	Generic structuring of data; integrated multimedia streams; distinction between content and presentation of data; semantic data; semantic classification of content; virtual re presentation of real-world items; virtual/augmented reality; cyber representation of active entities.
Data & Content Management	Technologies for managing and retrieving content while ensuring data integrity in dispersed and heterogeneous environments.	Guaranteeing the integrity of data, intellectual property management and protection; certification of content; offering unique virtual identity capabilities and management of the context; personal or professional content management and intelligent search.

Major challenges for Content technologies / Table ES-1

The volume of digital content is growing rapidly. We are entering the era of high-quality, high-definition multimedia streams – the most challenging and resource-demanding kind of content we know. Overall, there are two reasons for this explosion of content: first, the infrastructure for capturing content is spreading [see Digital Sensory System]; second, storage is available at very low cost, while accessible bandwidth is continuously increasing.

Content is available at different levels of abstraction, i.e. signal, data, information, and knowledge. It should be easily accessible and needs to be managed with sophisticated technologies (e.g. efficient searching and consistency management) in distributed and heterogeneous environments.

Moreover, having more and more data and information in a digital format makes possible not only easier storage, but also advanced manipulation, analysis and automated feedback. This creates opportunities for new applications and services, especially when combined with advances in network technology. One important issue is converting raw data into information and making it available to a variety of applications and services by aggregation or integration from different sources.

The key to coping with this challenge is thought to lie with context awareness and meta-data, which guide the user through the massive amount of content that can be exploited by semi-automatic reasoning on a formally defined semantic basis (through a semantic web and ontologies). This supports the conclusion that content without context or meta-data is of little or no use, especially if sustainable content management is the aim.

Another major challenge that is being addressed – one, which will become increasingly important in the years ahead – is the security of content (i.e. ensuring that data is received only by those authorised to receive it) and also its trustworthiness (i.e. ensuring that people receive the correct data). Key in this area will be ease of use and acceptance of the technologies by end-users.

INFRASTRUCTURES & BASIC SERVICES

A solid, trusted networking and computing structure is essential for providing ubiquitous services. Such an infrastructure consists of protocols, transport mechanisms and basic services. The network needs to be managed (either in a traditional way or self-managed – e.g. through ad-hoc networks) and maintained, and network services middleware should make it transparent to the applications that are deployed.

To achieve this, all kinds of resources such as processors, bandwidth, display, time, energy, memory and storage, network resources (routers, proxies, etc.) need to be managed as well. The network infrastructure is moving on from its role as a mere infrastructure to one that provides network services, i.e. middleware that delivers virtualisation to networked, distributed applications such as accounting, storage and profiling.

One important set of services involves security, property management and privacy issues, which play a role in several places of the digital world, such as secure transport, authorised access, and conditional access etc.

There are a number of issues related to the use of technology for 'shared' services (e.g. quality of life, publicly used infrastructures, etc.). These touch on technological challenges such as:

- collection of large amounts of data (e.g. home-based health services with information distributed to doctors, pollution control)
- large-scale distributed modelling (such as those required by e.g. complex sensor fusion or weather models)
- large-scale computations
- context awareness.

Network technology is expected to evolve rapidly. Such technology will be necessary to provide ambient intelligence, combined with capabilities for seamless distributed interoperable networking. The way towards ubiquitous access and self-organising networks will be paved by wireless networks, increased bandwidth and

IBS categories	Category definition	Major challenges
Network Transport	Technologies carrying digital data from one place to another.	Heterogenous networks interoperability; increased bandwidth, range and mobility support; IP (Internet Protocol) in any device; optimised streaming and broadcasting; fully distributed environments.
Network Services	Technologies for managing the dynamically changing network infrastructure for roaming users and services.	Ambient intelligence; seamless distributed networking capabilities; service coordination; identity management and profiling network services; support for accountable events.
Resource Management	Implementation technologies that take account of resource constraints (physical, computing, time, spatial, radio frequency).	Small lightweight devices with long-lasting energy source; optimising between conflicting goals; dynamic management.
Security	Technologies that provide safe and secure access to data, user indentification, etc.	Creating secure network services, protecting privacy, protecting content and recognition of ownership of Intellectual Property, creating easy, reliable, safe personal identification.

Major challenges for Infrastructures & Basic Services technologies / Table ES-2

a reliable QoS Internet Protocol (which will be used by web services and P2P[6] protocols), service coordination, identity management and profiling services.

However, if an appliance, device or system is able to function optimally, constrained resources such as memory and storage, bandwidth, display size, time, power, and network resources will have to be properly managed. Terminal power management is critical to this. To share resources across organisations, complex distributed architectures will use new technologies such as grid technology. For critical applications, dynamic resource management is becoming more and more important (e.g. for self-healing or self-protecting).

In the background, the key challenge for security will be to get users to trust systems, giving them the assurance that “they are in control”. This is a major technical challenge, however, as the systems in use will become

[6] *Peer-to-Peer*

larger, more networked and more dynamic – whereby users may find them less predictable. To address this concern, security will become pervasive, and will be handled at all stages of the software life cycle.

HUMAN-SYSTEM INTERACTION

Software-intensive systems are becoming more and more complex and, at the same time, more and more people are using them. The interaction between the user and a system has a major impact on the acceptance and usefulness of these systems. The method of interaction depends on a number of factors, some of which themselves depend on the role of the user, the interaction device, the user's experience and their environment. A critical factor in the acceptance of information technology will be ease of access for non-technical users. Consumers in the mass market require easier interfaces than technology innovators and early adopters.

User interfaces need to support multilingual versions in distributed, collaborative, multicultural and multi-user environments, and must allow simple and quick authoring, navigation and access to multimedia data. Particularly important are multimodal (referring to modalities such as sound and vision), adaptive, personalised, scaleable user interfaces. New appliances, for example in intelligent Home and Nomadic devices and those used in industrial applications and systems (process control, air traffic control, medical systems etc.) require multimodal interaction.

In recent years, the games industry has also been particularly influential. Heavy competition and the demand for ever better versions in very short cycles have stimulated the development of Human-System Interaction (HSI), which comprises technologies that handle interaction with the user.

Even as people become increasingly familiar with complex systems, spending time coping with complexity is becoming less acceptable. HSIs should therefore hide underlying complexities from users and provide the best possible user experience, so that the user feels in control. A specific challenge here is to intelligently select appropriate modalities and provide "zero configurability" within multidimensional environments consisting of different networks and access technologies, devices, people and services. In addition, future user interfaces will need to be able to learn user preferences and to store, manage and disseminate this information to relevant related services and devices. This process needs to be secure and it must protect user privacy. However, acceptance of new HSIs that replace familiar, although complex, ones will take time unless the new HSIs can naturally support accepted modes of interaction.

In recent years, the various technologies have probably been overemphasised. For those who use them, it is the entire user experience that determines whether the new technologies provide real added value or not. It should also be noted that the added value usually comes indirectly through the services and applications that intelligently apply the new technologies – not from the technologies as such.

Therefore, in order to be able to fulfil users' needs and meet general requirements for future systems, the R&D in underlying technologies [see Chapter 2 on Content] and platforms [see Chapter 3 on Infrastructures & Basic Services] should be done in close cooperation with the R&D in user interface and interaction technologies [see Chapter 4 on Human-System Interaction], and vice versa. Moreover, other disciplines outside the area of software systems also need to be involved in the process so that software engineers can better understand user requirements, transfer them into software systems and improve the acceptability of outcomes; this includes, among others, behavioural science specialists and physiologists.

In short, the target for the new HSIs is that they should be:

- simple, self-explanatory and easy to use
- intelligent, context-aware and adaptive
- seamless and interoperable.

The whole process should largely be user-driven, which also poses new demands for software and systems design and engineering [see Chapter 5 on Engineering technologies and challenges].

ENGINEERING

Software-intensive systems are programmable systems based on one or more integrated controllers. Even today, less than 10% of the controllers produced are built into computers. Most of them can be found in systems such as cars, phones, washing machines, aircraft, robots, traffic systems, cameras and audio equipment. And this trend towards programmable systems is accelerating. Software not only increases the variability, configurability, extendibility and changeability of everyday systems, it will also soon allow for a greater variety of functions based on the advanced information processing capabilities built into these systems.

Future systems will be even more dominated by software. As a consequence, the engineering and maintenance process of software-intensive systems is undergoing dramatic changes because software is becoming not only a major part of the product but also a central aspect of system engineering.

Embedding software into systems increases the complexity of these systems as well as the complexity of the engineering process. Complexity becomes apparent whenever it becomes difficult to comprehend and manage all the aspects, requirements, consequences, interrelationships and relations associated with a specific product and the product creation process. It emerges from the combination of architecture decisions, restrictions on existing physical resources, integration of legacy functionality, required non-functional properties and the use of heterogeneous technologies.

In general, complexity increases the effort needed to develop products, services or infrastructure and increases the already high tension between time-to-market and general development cost on the one hand and the quality and adequacy of the product on the other. The effective engineering of efficient, reliable and safe systems is essential and needs support from appropriate technologies, such as methodologies, notation languages, design and implementation techniques, generation techniques, tools, knowledge, processes, guidelines and maintenance support.

Although the integration of hardware and software will present system engineers with new challenges, many of the engineering problems mentioned above are mostly software-related. This is partly due to the focus of this Roadmap but it also reflects the fact that software engineering is still a very young discipline.

Progress in software engineering support will depend on advances in methodologies, models, tools and implementation techniques that support the efficient implementation of embedded systems and services. Engineering technology will require techniques for mastering complexity and capturing high-level specifications of distributed systems. It will also require implementation tools that, firstly, support short time-to-market development cycles, and secondly, guarantee not only cost-efficient, resource-limited implementation on a wide variety of platforms, but also design flows that supports the development process in multi-disciplinary teams.

Engineering categories	Category definition	Major challenges
System Engineering	Techniques, methodologies and tools for the design and construction systems under constraints (time-to-market, technological, legal, economic and legacy).	Evolutionary systems; product line engineering; automation in verification and validation; system architecture trade-off analysis; hardware/software co-design.
Software Engineering	Techniques, methodologies and tools for the design and construction of architectures and adaptive technologies for implementation, deployment, execution, exploitation and maintenance of software systems.	Component markets and software suppliers; cross-cutting concern engineering; design pattern support; model-based development; self-organising software agents.
Engineering Process Support	Methodologies, techniques and tools that support an engineering and distributed engineering process.	Integration and interoperation of engineering tools; distributed and collaborative engineering; configurable methodologies and process standards; requirements-driven process management; knowledge-based engineering.

Major challenges for Engineering technologies / Table ES-3

However, we do not need totally new technologies for all problems. Experience has shown that integration, harmonisation and further development of existing ideas, techniques and corresponding tools can deliver significant progress. Moreover, software development is still often viewed and treated as an art rather than a mature engineering discipline that requires engineers to follow well-defined rules and guidelines. Progress can also be expected from the global exchange of best practice engineering experiences.

But it is not just technological challenges that we should be expecting. To a large extent, success will also depend on other factors, such as appropriate business models (e.g. for components), the way we deal with intellectual property for software, and the way in which the growing open source development scene influences the engineering of systems. To take just one example, the move to component-based systems will certainly impact the business model of industrial organisations and software technology and tool vendors. Component-based architectures will lead to sharing and exchange of common components between various applications. This will enable the development of new products and new technologies through the assembly of existing components.

It is still unclear how end-users will finally pay for components. Business models may range from “free” software and open source models, to licensing components and pay-per-use. With respect to the current situation, it is not inconceivable that most standard components will emerge from open source development, which may well prevent a classic component market appearing.

A simplified conclusion of the findings of this Roadmap

In years to come, *software-intensive systems* will incorporate four basic features (compared to their current status):

- They will be dynamic evolutionary systems.
- They will exhibit adaptive and anticipatory behaviour.
- They will process knowledge and not only data.
- They will allow the user to stay in control.

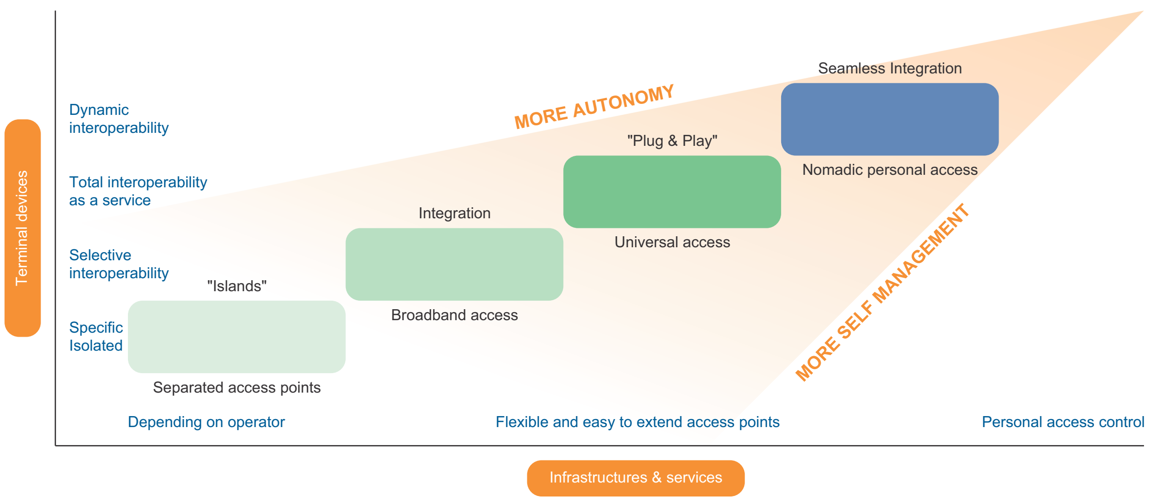
There are two kinds of keys to the development and deployment of these systems:

- Key drivers for *acceptance* are
  - interoperability of products, systems and applications
  - the "-ilities" – security, usability, testability, reliability – and safety

- Key issues for *implementation* are
  - mostly technical: mastering size, complexity and adaptiveness
  - mostly economic: middleware business models and costs

At the crossroads of technology and business: the challenge for the years ahead

Our work on this second edition of the ITEA Roadmap for Software-Intensive Systems has confirmed most of the findings of the first. It has been possible to validate the gradual evolution that we then described: *"Our technological environment will become, step by step, more and more networked and more and more autonomous and self organising […] From now on to this long-term vision, generally two stages may be foreseen […]: the self configurability and interoperability of the terminal devices and the integration of the different networks."* [7] The evolution of applications during the past three years has confirmed the projections made in the previous document [see Figure ES-1].



The road towards convergence / Figure ES-1

[7] *From page 4, Technology Roadmap on Software Intensive Systems, ITEA, March 2001*

These three years have also shed more light on some very important features related to software-intensive systems that will shape the fundamental challenges that the industrial community must address in the next few years.

The first feature concerns the very special nature of software (high cost of development and very low cost of reproduction) and its ubiquity: it pervades all kind of applications. The second is global worldwide networking with all systems collaborating with one another. Combined, these two characteristics present us with challenges which were not yet fully apparent in the first edition of the Roadmap, ones in which it is impossible to separate business, industrial and technical considerations. These can be stated in a very few words – but that does not make them easy to implement.

All parts of the network are entering a state of *continuous change*, with respect both to the services they provide, and to their number and capabilities. This is having a cascade effect, whereby changes made in any one part of the system may lead to the need for changes in other parts. Here, legacy is the key issue, one whose consequences will usually be unknown until they are tackled. To prevent chaos, we need brand-new ways of creating software systems that are self-configuring, self-adapting and self-healing, based on loosely coupled, independently evolving components from numerous suppliers around the world.

The very *nature* of what we are building has changed. We were trained to build rock-solid software-powered pyramids or cathedrals, with well-defined traditional, respectable organisations, employing one architect and one master contractor. We now have to deploy and use a thin web of players who are dynamically interconnected, extremely diverse, unrelated – and sometimes antagonistic. How can we combine the agile and cooperative development we need with four other vital factors: respect for Intellectual Property Rights, the acceptance and development of open standards, the acknowledgement of liability, and rapid access to the marketplace? This is where lessons from the Open Source Software movement may be useful as well.[8]

As we've proved over the last years, ITEA is actively taking up the gauntlet.

[8] *See the ITEA Report on Open Source Software, January 2004 (downloadable from www.itea-office.org)*

ITEA Technology Roadmap

for Software-Intensive Systems

ITEA Office

Eindhoven University of Technology Campus
Laplace Building 0.04
PO Box 513
5600 MB Eindhoven
The Netherlands
Tel : +31 40 247 5590
Fax : +31 40 247 5595
Email : itea@itea-office.org
Web : www.itea-office.org

ITEA
INFORMATION TECHNOLOGY
FOR EUROPEAN ADVANCEMENT

ITEA is a EUREKA strategic cluster programme

Σ! 2023