

ON THE ORIGIN OF EVOLVABLE SYSTEMS

EVOLVABILITY OR EXTINCTION*

P. Daniel Borches

Department of Engineering Design, Laboratory of Design, Production and Management
University of Twente, The Netherlands
p.d.borches@ctw.utwente.nl

G. Maarten Bonnema

Department of Engineering Design, Laboratory of Design, Production and Management
University of Twente, The Netherlands
g.m.bonnema@ctw.utwente.nl

ABSTRACT

System evolvability is vital for the industry for the survival of complex systems. It is however difficult to achieve as this property is not well understood. Therefore there are no formal means of how to Design for Evolvability (DfE). Also there are no applicable ways of assessing evolvability, so it can not be proven whether a system or system design is more evolvable than another. This paper reviews previous research done concerning system evolvability, marking the field of our future research by proposing a reviewed definition of system evolvability and adding new concepts such as the key drivers of system evolvability. The objective is to find a means to steer the design process in a direction so that evolvability is designed into the system. System evolvability of complex systems is studied by using the Magnetic Resonance Imaging (MRI) scanner developed by Philips Medical Systems as a study case.

KEYWORDS

Evolvability, system evolvability, evolvability key drivers, assessing evolvability, design for evolvability

1. INTRODUCTION

Evolvability is a property that is vital in the industry sector for the survival of systems (Isaac and McConaughy 1994; Steiner 1998). The effort and resources required to adapt to changing requirements or different environments in complex systems can be significant. Even minor top-level functional changes can have lengthy, costly and difficult to predict development cycles. An evolvable system is best suited to cope with those problems. Therefore, evolvability should be taken into account during the

design process of the system. However, nowadays there is still no clear way to know whether a complex system is evolvable or not. Let alone how to design for evolvability.

Well-known complex systems, such as the TV and the Internet, evolved from rather primitive systems with limited scope into advanced global systems. Unlike other systems, e.g. the compact cassette recorder, they were able to evolve. However, in several aspects these systems still suffer from choices made at design time of the systems. It can be argued that any system can evolve. However, the resources and cost to evolve a system not designed to evolve can be even bigger than to redesign the system from scratch (e.g. knocking down a building might be cheaper than renewing it to meet current standards).

Nowadays, designing a system that is evolvable is considered best practice in many industrial domains due to the benefits it can provide (Schulz, Fricke et al. 2000). Life-cycle cost is reduced by a long-lived architecture that eases evolution rather than large-scale system redesign. Evolvability gives additional flexibility, as the company can either reuse existing infrastructure to tackle changing requirements or develop a new product. This reduces the time to market and increases market share while adding value to the customer. By designing the system for evolvability, it will be better suited to cope with unknown future requirements.

Despite the apparent need for system evolvability on complex systems, it is still both difficult to understand and difficult to achieve. It is difficult to understand, as there is not enough research done in the field. Moreover, system evolvability can be confused with other similar system properties in literature such as adaptability, changeability, flexibility, extensibility, and enhanceability.

*This work has been carried out as a part of the DARWIN project at Philips Medical Systems under the responsibility of the Embedded Systems Institute. This project is partially supported by the Dutch Ministry of Economics Affairs under the BSIK program. 1

Even having a good understanding of system evolvability, it is still difficult to design for evolvability. As it is difficult to estimate the future, it is not easy to design a system able to evolve in order to adapt to unknown future requirements [reference]. It also requires foreseeing how possible changes will propagate through the system. Designing-in evolvability is even more difficult when dealing with new products and features, as designers have then limited experience in the system and its environment. Our objective is to find a means to steer the design process in a direction so that evolvability is designed into the system: Design for Evolvability (DfE).

In this paper, a literature review regarding system evolvability is presented. System evolvability is analyzed from the industrial point of view using the industry as a laboratory for our research. A complex system -Philips Magnetic Resonance Imaging scanner (MRI)- will be used to understand evolvability on complex systems. The following section reviews previous work regarding system evolvability. Section 3 focuses on evolvability of complex systems by describing the relevance of system evolvability for the industry, the key drivers of system evolvability and ways of assessing evolvability. Section 4 describes the study case that will be used in our research regarding system evolvability -Philips MRI scanner-. In the last section the conclusions and future work are presented.

2. EVOLVABILITY REVIEW

Many years have passed since the first paper regarding ‘system evolvability’ was published (Isaac and McConaughy 1994); some theoretical work has been done in the field (Rowe and Leaney 1997; Rowe and Leaney 1998; Christian III 2004); a few papers attempted to measure evolvability of complex systems (Christian III 2004; Christian III and Olds 2005), and few more tried to analyze evolvability on real systems (Rowe and Leaney 1998; Christian III and Olds 2005). The importance of adopting evolvability has been discussed by several authors (Isaac and McConaughy 1994; Rowe and Leaney 1997; Ring and Fricke 1998; Steiner 1998; Christian III and Olds 2005) and its role in the system architecture has been described in (Isaac and McConaughy 1994; Steiner 1998).

System evolvability is still, however, almost an unexplored field. System’s evolvability definition is still open for discussion (Rowe and Leaney 1997; Rowe and Leaney 1998; Christian III 2004; Christian

III and Olds 2005); the key drivers of evolvability are not clear; there is no formal way to assess or measure evolvability (Christian III 2004; Christian III and Olds 2005); evolvability can be confused with other system’s properties such as changeability (Fricke and Schulz 2005) and there is almost no real work done in the field. In addition, there is no formal research regarding ‘design for evolvability’. In this section we briefly review previous work regarding system evolvability.

2.1. Evolvability in literature

Evolvability as a concept has its deepest roots in the biological and social sciences; most of the literature regarding evolvability refers to those sciences. Darwin’s theory, characterized by heritable variation and natural selection, is often used as a starting point. For example, in biology evolvability has been variously defined as ‘*the ability of a population to produce variants fitter than any yet existing*’ (Altenberg 1994) or ‘*the genome’s ability to produce adaptive variants when acted on by the genetic system*’ (Wagner and Altenberg 1996) and so on. Most of those biological and social definitions are however not appropriate for complex man-made systems as species and systems are not similar enough (e.g. there is no clear analogue of ‘gene’). In other fields of research such as software, evolvability is defined as ‘*the capacity of software products to be evolved to continue to serve its customer in a cost effective way*’ (Cook, Ji et al. 2000). Software definitions trend to be too specific regarding software properties and attributes. Although system and software share some common properties, as software definitions are domain specific, they cannot be used. Therefore, an appropriate definition for system evolvability is needed.

Over the years, there have been a few attempts to define system evolvability. Despite these attempts, there is still no broadly used definition of system evolvability. Some of these definitions are:

- ‘*System evolvability is a trait of a system that allows the system to be easily modified due to changes in the environment*’. (Percivall 1994)
- ‘*System evolvability is a system’s ability to withstand changes in its requirements, environment and implementation technologies*’. (Rowe and Leaney 1997)

- System evolvability; *‘an attribute that bears on the ability of a system to accommodate change in its requirements throughout the system’s lifespan with the least possible cost while maintaining architectural integrity’*. (Rowe and Leaney 1998)
- System evolvability; *‘the capacity of a system to successfully adapt to changing requirements [IEEE, 1990] throughout its lifecycle without compromising architectural integrity. Furthermore, an evolvable system must meet the new needs of the customer in a more cost effective manner than developing a new system’*. (Christian III 2004)

Each definition tries to identify the main factors that evolvability must deal with, such as changing requirements and environment; and how it should be addressed; easily, in a cost-effective manner and so on. However they do not agree upon those factors and how they should be addressed. Other definitions of system evolvability are similar to those presented.

2.2. Assessing evolvability in literature

Previous approaches used to assess evolvability are based primarily on the work of Mario Bunge (Bunge 1977). Although philosophical in nature, Bunge’s work has led to extensive research in ontology’s application to the engineering and computer science disciplines.

Regarding system evolvability, Rowe and Leaney proposed a model of systems architecture evolution based on an ontological basis (Rowe and Leaney 1997). In this model, as evolution is considered a type of change, it is modeled as an ‘event’. An event is a pair of states, where each state (start and end) exists in the ‘possible state space’. This concept of modeling evolvability implies that both the initial and final evolved states must be known. John A. Christian III tried to use Rowe and Leaney’s approach to measure system evolvability on real space systems, by adding some metrics such as figures of merit (Christian III 2004). However, in later publications the approach was replaced with another approach based on experts’ estimations regarding evolvability (Christian III and Olds 2005). These estimations were converted into a numerical fashion and processed to obtain data. This method relies on a self-made scale of evolvability to provide numbers and the experts’ understanding of evolvability to select the appropriate numbers. It provides a structured framework in which the metrics

may be discussed and measured, still, the methodology is mostly based on expert’s opinion instead of qualities or attributes of the real system.

3. EVOLVABILITY OF COMPLEX SYSTEMS

Almost every system is inherently complex. In a complex system, knowledge of the elementary building blocks does not give a glimpse of the behavior of the global system. We accept that processes that can occur simultaneously on different scales or levels are important, and the intricate behaviour of the whole system depends on its units in a non-trivial way. It is to say, systems are complex when they are in practical sense unpredictable, uncertain, etc. Complexity has been widely studied by several authors and is out of the scope of this paper. As we are interested in the industrial sector, in our research concerning evolvability of complex systems we focus on those complex systems that have the following characteristics:

- man-made commercial products
- multidisciplinary systems (the knowledge and disciplines of the system are too large to be managed by one or a small group of persons)
- software-intensive.

We believe that these characteristics are the most relevant for the industry. A few examples are airplanes, automobiles, satellites and MRI scanners.

3.1. Evolvability in the industry

Nowadays, designing a system that is evolvable is considered best practice in many industry domains due to the benefits it can provide (Schulz, Fricke et al. 2000). With evolvable systems, companies can benefit from a system that can adapt to changing requirements or different environments at a cost less than is required to build a new system. Modernization of evolvable systems is expected to take less time and reduce costs. Evolvability enables easier insertion of new technology and mitigation of the risk of obsolescence in the system. Life-cycle cost is reduced by a long-lived architecture that eases evolution rather than large-scale system redesign. In addition, evolvability gives additional flexibility, as the company can either reuse existing infrastructure to tackle changing requirements or develop a new product.

In most industrial domains, the current way of working is based on incremental design and incremental development. The systems are not designed from scratch but reuse existing designs and artifacts. Designing a system for evolvability simplifies this incremental way of working and ensures easy adaptation to changing requirements. This reduces the time to market and increases market share while adding value to the customer. By designing the system for evolvability, it will be better suited to cope with unknown future requirements.

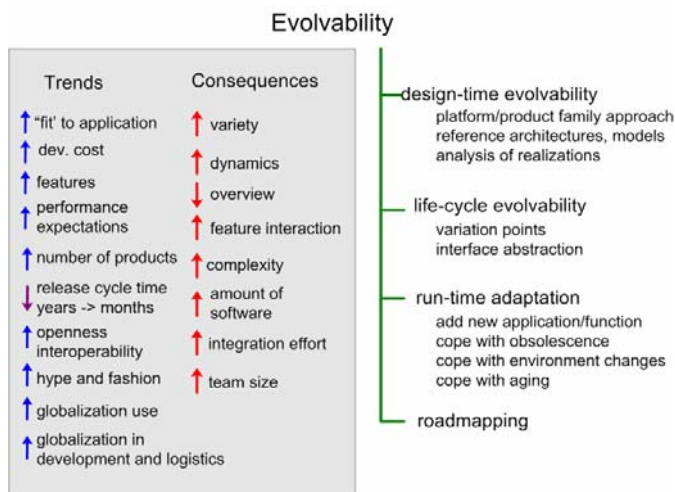


Figure 1 Market trends causing evolvability and consequences (courtesy of Gerrit Muller)

As shown in the Figure 1, current trends in the industry and their consequences, causes evolvability to be required in the industry at different levels. Among these trends and consequences, it is worth to mention:

- *Complexity increase:* solutions in response to demands for new features to add value to the system such as higher performance, greater ease of use, improved reliability, more safety and interoperability have been increasing rapidly in recent years and are driving an increase in system complexity.
- *Time to market pressure:* the innovation cycle of products is decreasing. In most industrial sectors, competition is fierce and price erosion is fast. In a global market, product manufacturers are under severe pressure to reduce product and development costs and yet remain technologically one step ahead of the competition.
- *Open system demands:* customer's expectations are that their products can

connect to a variety of products and systems from different manufacturers. To support this interoperability, the ability to communicate and interface using open standards is essential.

- *Product families:* customers demand personalized products. The final system is not just one product but a family of products. This leads to an increase in the effort required during the life-cycle design, as product families are derived from a common platform in order to enable re-use of current knowledge and infrastructures.

As stated before, an evolvable system can help to cope with these trends and their consequences. Some industrial sectors such as aerospace, automotive, naval and DoD, have already identified the necessity of adopting evolvability in their systems and have placed it as a primary requirement (Steiner 1998; Schulz and Fricke 1999).

3.2. System evolvability key drivers

To better understand system evolvability it is important to understand the attributes and properties that determine whether a system is or not evolvable; the **key drivers** of evolvability of complex systems. We use here the term 'key driver' in a broad sense, in a high level of abstraction, as the key drivers of system evolvability also depend on the kind of system. Once the key drivers of system's evolvability are well-defined, it will be easier to know what is relevant for the system to be evolvable.

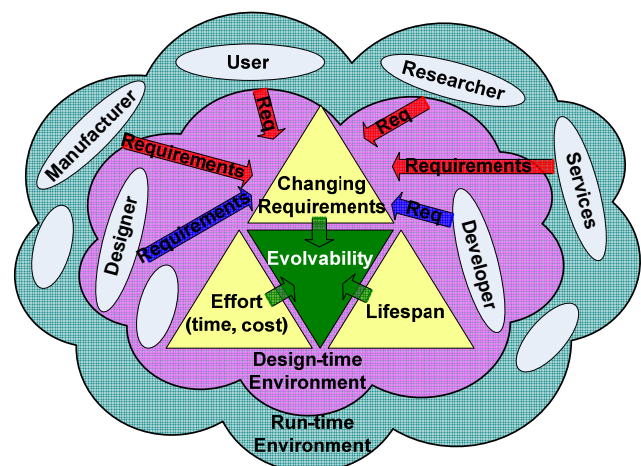


Figure 2 System evolvability key drivers

Based on previous work and previous definitions, we have identified the main factors of system evolvability: *changing requirements, environment,*

effort and *lifespan*. As stated before, maybe there are more key drivers, as it also depends on the kind of system; however, we think that these key drivers are common to all complex systems. Consequently, they need to be taken into account when designing the system. As shown in the Figure 2, system evolvability is affected by:

- *Changing requirements*: it is clear that for a system to be evolvable, it must cope with changes in the set of requirements. All complex systems will face changing requirements through their lifespan. The source of those changing requirements is typically the stakeholders. The changes can be *internal* (e.g. a designer proposes a change in the implementation technology) or *external* (e.g. user demands new functionalities). Several kinds of changes can occur, such as functional, behavioral, physical, performance, etc; each of them with diverse impact on the system. The ‘frequency’ and ‘impact’ of those changes has an affect in the evolvability of the system. A system with frequent and strong impacts has fewer opportunities to survive, as it has less time and requires more effort to properly adapt to those changes.
- *Environment*: plays an important role in the evolution of the system, as the path of the evolution will be determined by the environment of the system. Different environments lead to different requirements and have different impact on the system. This is illustrated in nature by species developing different capabilities, depending on their environment.

We differentiate two kind of environments; environment at *run-time*, and environment at *design-time*. The run-time environment is the one the system faces once it is deployed, and it is determined basically by the ‘external’ stakeholders’ behaviour and their environment. For example, the behaviour of the system may vary depending on the location (different pressure, temperature, etc); the requirements to meet can be different depending on the country (the same TV has different requirements in USA than in Europe such as the power supply, the user interface, etc), and so on. The design-time environment is the one the system faces at

the company where it is being designed. It is determined by the ‘internal’ stakeholders and their environment. For example, in the design-time environment, aspects such as the company philosophy and company behaviour can have an affect on the evolvability of the system.

- *Effort* (time and cost): even minor top-level functional changes can have lengthy, costly and difficult to predict development cycles. In order to be evolvable, a system should adapt with less effort than required for designing a new system. The effort required to adapt the system determines whether the system is suited to evolve, or is better to be replaced by a newer system. Two factors determine the effort to evolve the system, *time* and *cost*. The former determines whether the system is going to evolve on time and the second one whether the resources required are worth evolving it. The less effort the easier the system adapts; the more evolvable the system is. A large evolution cost or evolution time would probably force the system to become ‘extinct’.
- *Lifespan*: evolvability is not equally desirable for all systems, such as those with sort lifespan, insensitive to change over time or belonging to a slowly changing market. It is highly desirable for systems with long lifespan, highly interconnected systems with need for future growth or those systems requiring a large degree of infrastructure support. These systems require to adapt to changes throughout their lifespan.

Two factors play an important role in the lifespan of the system; *market lifetime* and the *maturity* of the system. The shorter the market time of the system -‘throw-away’ systems such as mobile phones-, the faster the system design needs to evolve. Mature systems are more likely to be able to evolve, as there is more experience and knowledge about them. However, sometimes old systems suffer from decisions made at their design-time, affecting their ability to evolve. Also, the implementations may be based on end-of-life or obsolete technologies (Herald Jr., Verma et al. 2007).

3.3. Evolvability definition

At this point, we consider that a revision of the evolvability definition is needed, in order to use it as the starting point for our future research. Part of the problem in focusing on evolvability explicitly during design is the manifold ways in which it has been defined. Previous definitions of system evolvability - see Section 2.1- only deal partially with the evolvability key drivers. The first definition (Percivall 1994) is, from our point of view, incomplete; as it does not take into account changes in requirements in the system evolvability definition. The second definition (Rowe and Leaney 1997) seems more accurate as it introduces the changes in requirements and implementation technologies as part of the definition, however, instead of 'withstand' we believe that the system should 'adapt' to changes; and it should be done in an efficient way. The third definition (Rowe and Leaney 1998) introduces cost and lifespan as part of the definition; however, it only takes into account changes in requirements as part of the evolvability definition. The same applies to the last definition (Christian III 2004).

Based on all the points previously discussed and in our own expertise, we propose the following definition for system evolvability:

Evolvability is the *system's ability to adapt to changing requirements and different environments throughout its lifespan in a time-efficient and cost-efficient way.*

Note that the proposed definition modifies the previous definitions presented by taking into account all the evolvability key drivers; changing requirements, different environments, lifespan and effort (time and cost), and how it should be addressed; in a time and cost-efficient way.

3.4. Evolvability as a composite of properties

As stated before, in literature many terms refer to a system's ability to accommodate change. It is worth to analyze some of these terms to easier understand their connection with evolvability:

- *Adaptability*: the ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed (Rowe and Leaney 1998).

- *Flexibility*: the property of a system to be changes easily and without undesired effects (Schulz and Fricke 1999).
- *Changeability*: the ability to meet changing situations and diversified operations with minimum disruption or delay (McCay 1996).
- *Extensibility*: the capability of being extended resulting in easier, faster, and less costly upgrade in capability (Bensley, Fisher et al. 1995).
- *Enhanceability*: the ease with which new functionality can be added to a system (Dasgupta 1991).

Other system properties deal with change, such as; *portability*, being able to change the underlying platform; *upgradeability*, the capacity of upgrading the entire part of the system with improved features; *extendability*, the capacity to add options or new features; and *maintainability*, the capacity of maintaining the well-being of the system (Muller 2004). Though this list is not exhaustive and there are more definitions of those properties, it is significant enough to demonstrate that many different terms deal with different aspects of change. However, even if a system can accommodate change one way or another, it does not automatically mean that it is evolvable. A well-designed system may be able to adapt to some changing requirements or different environments, but it may be unable to evolve to adapt to unknown future requirements because the effort required is too high.

As described before, there are many ways in which a system can accommodate change; however they are related to specific forms of change. On the other hand, evolvability deals with adapting change in all possible ways. It can be argued therefore, that evolvability is, in a broad sense, a composite of the other system properties that deal with adapting to change.

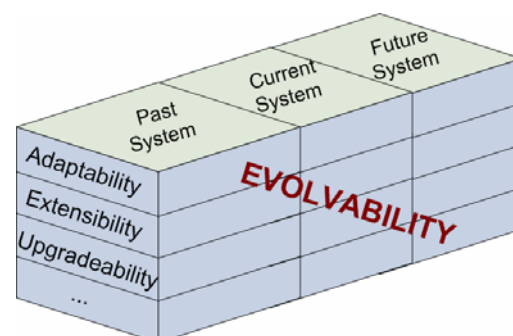


Figure 3 Evolvability as a composite of properties

Yet, evolvability is not just the composite of other properties that deal with accommodating change; as shown in Figure 3, evolvability introduces a new dimension to the context: time. All the other properties focus mostly on the current system to see whether or not they can accommodate change one way or another. Evolvability refers to how the system or system design changes from one generation of a product design to the next, such as specifying which elements of the design are passed down and which elements of the design are new to the latest generation. It is to say, evolvability deals not only with the current system, it also deals with the future system. Evolvability enables the future system to easily adapt to changing requirements or different environment, with less effort than redesigning the system, which can be achieved through flexibility, extensibility, upgradeability, or other property.

3.5. Design for Evolvability (DfE)

Designing a system able to survive any (hostile) environment and able to adapt to all change -if that is possible- would be unnecessarily expensive. With evolvable systems, system designers recognize that the system will have to face unpredicted situations, and try to minimize their effect when they happen. Some changes in the requirements can be anticipated, however, unpredicted changes will occur since estimating the future and the consequences for a system is not a science (De Neufville 2004; Smaling 2005). Given the need for evolvability, what is next needed is a mechanism to provide it. However, nowadays there are no formal means of designing for evolvability and it is usually delegated to the designer's intuition.

As stated before, our objective is to find a means to steer the design process in a direction so that evolvability is designed into the system: Design for Evolvability (DfE). This strategy aims to enable a company to be responsive to changing requirements (e.g. change in customer needs) or environments (e.g. changes in the market) by implementing the necessary evolvability patterns throughout the entire life-cycle of the system.

Overview

It is well-known that late changes in the life-cycle of the system can cost more than the same changes made earlier. In an effort to eliminate these costly changes, development processes focus on refining the requirements and design by choosing the *exact* requirements and the *correct* design (Isaac and

McConaughy 1994). Thus, we contend that system evolvability is best addressed in early stages of the design process at the system level, as this level best allows to model high-level requirements.

An evolvable system should be built on the aspects of the system which are most likely to remain unchanged (Isaac and McConaughy 1994). These 'islands of architectural stability' (Percivall 1994), enable a complex system to evolve much quicker and easier than what might otherwise be possible. By defining independent functional modules, each of them can evolve at a different pace, without affecting other modules. This is also termed as *functional isolation* (Christian III 2004). Those modules should be identified by the system architect as early as possible in the development process.

To achieve the development of evolvable systems, DfE will focus on the creation of an *evolvable architecture*, the description of an *evolutionary development* process and establishing an *evolutionary environment*.

Evolvable architecture

In order to be evolvable, a system should have an 'evolvable architecture' that drives an evolvable design. The evolvable architecture is meant to provide the designers with the 'tool' needed to implement changes and understand the system and the impact of changes. While implementations of the system may be changed several times during the lifespan of the system, it may be only the architecture that remains from the original system.

Although the concept of 'system architecture', what should be included in it and its relation with system properties such as evolvability is still unclear (Steiner 1998), its utility, especially in evolvability is clear (Muller 2008). The system architecture provides the framework in which the design is performed. An evolvable architecture should be developed and documented including the goal of evolvability and the key characteristics of the system relative to evolution that the system is likely to experience over time. In addition, the context and the life-cycle of the system as well as the design should be part of the architecture.

It should be noticed however, that whether a system persists or ceased to be used depends on a kind of selection of functional and non-functional properties that cannot be all captured in the architecture (e.g. a mobile phone may be successful because of battery

efficiency or because it looks impressive and attracts customers).

Evolutionary development

In order to support evolvability, certain additions to the development process are required. To ensure the development of an evolvable system, potential areas of change should be accommodated in the current requirements, design and implementation. The process to ensure the development of an evolvable system is called *evolutionary development* (Isaac and McConaughy 1994).

The design activities of the evolutionary development should support the creation of an 'evolvable architecture'. As stated before, the concept of system architecture and its relation with system properties such as evolvability is unclear, however in order to achieve an evolvable system, the system architecture should include architectural constraints and design guidelines that enable evolvability. Components and interfaces must be developed to satisfy not just the functional requirements, but the evolutionary requirements. Implementation and test activities must also verify that evolvability requirements are met. Finally, the engineering process or the system should provide feedback mechanisms both in the design and operational phases to early identify and adapt changes.

Evolutionary environment

The success in re-architecting the system for other system 'ilities' (Richards, Hastings et al. 2007) illustrates the importance of considering methods that extend beyond the domain of physical design to include organizations and operational behaviour. Thus, not only delivered systems or system designs have to incorporate evolvability attributes. Companies themselves need to incorporate evolvability concepts within their entire development process, it is to say, an evolutionary environment. A success in the transition from a non-evolvable system to an evolvable system (architecture) even without major physical or functional modifications might be possible by setting up the appropriate (design-time) environment.

4. CASE STUDY: PHILIPS MEDICAL SYSTEM MAGNETIC RESONANCE IMAGING (MRI) SYSTEM

To study evolvability of complex systems, we use the industry as the laboratory for our research. We have chosen the industrial sector of high-end medical systems, specifically that of MRI scanners. MRI scanner technology is relatively young, is increasing in popularity as products become more powerful and effective for clinical diagnosis and therapy in many areas. Although the MRI technology is relatively young, it has evolved rapidly to adapt to new technologies and requirements.

Darwin project

The research on design for evolvability of complex systems is carried out by the **Twente University** in the **Darwin project**. Darwin is a consortium of industrial and academic partners that has been set up to carry out the project with the Embedded Systems Institute (ESI). The partners are Philips Medical Systems - MR division (Carrying Industrial Partner), Philips Research, Twente University, Delft University of Technology, Eindhoven University of Technology, University of Groningen, and the Free University of Amsterdam. The goal of Darwin is to understand evolvability as a system property; to identify, create, and apply constructs, models, and methods to support evolvability; to support the tradeoff decisions the architect will have to make with respect to evolvability; and to support the sub-system and technology lifecycle view of a system. The Darwin project is executed using the industry-as-laboratory paradigm. Hence, the researchers are working closely together with designers, developers and architects of Philips Medical Systems (Laar, Loo et al. 2007).

Company background

Koninklijke Philips Electronics N.V. (Royal Philips Electronics N.V.), usually known as **Philips**, is one of the largest electronics companies in the world, founded and headquartered in the Netherlands. In 2006, its sales were €27 billion and it employed 121,700 people in more than 60 countries (Medical 2007).

Philips Medical Systems is a global leader in diagnostic imaging systems, healthcare information technology solutions, patient monitoring and cardiac devices. Philips also provides customer services such as financing, consultancy and maintenance & repair. Their product line includes technologies in X-ray, ultrasound, magnetic resonance, computed tomography, nuclear medicine, positron emission tomography, radiation oncology systems, patient

monitoring, information management and resuscitation products.

Magnetic resonance imaging (MRI)

MRI is an imaging technique used primarily in medical settings to produce high-quality images of the inside of the human body. MRI is based on the principles of nuclear magnetic resonance (NMR). In 1971, research showed that the magnetic relaxation times of different tissues differs, enabling magnetic resonance for scanning the inside of the human body. It was during the late 1960s that Philips started to conduct their own MRI research and they produced the worlds first head images using the MR principle in 1972.



Figure 4 Philips Achieva MRI scanner (Photo courtesy of Philips Medical Systems)

An MRI scanner has a large electromagnet around a human-sized tube (bore) in which the patient or test subject lies -see Figure 4-. The electromagnet consists of coiled wires made of super-conductive material. The gradient coils are used to vary the strength of the main magnetic field. Radio frequency coils that transmit and receive radio signals are used in combination with the magnetic field to induce and measure a resonant field within the patient or test subject. The patient's protons align themselves in the same direction as the magnetic field applied by the scanner. Under the control of the operator, the radio frequency coils in the scanner emit short bursts of radio waves that cause the protons to re-align. When the radio waves and the protons vibrate at the same frequency, the protons absorb some of the radio wave energy. This is called resonance, and it is from this phenomenon that the term magnetic resonance is derived. Each time that the radio frequency coils are turned off, the protons go through a process of returning to their initial orientation. As the protons do so, they emit energy. This energy generates a voltage in a receiving wire antenna and is then converted into a digital signal that serves as the basis

for MR images. Different tissues give off signals of different strength depending on their chemical composition and location (Hornak 2003).

MRI system architecture overview

The MRI is based on several building blocks termed chains. A chain is a hierarchically organized, functional unit of the MRI system. It may consist of a number of hierarchically lower building blocks and has a unique name and description with a tree structure.

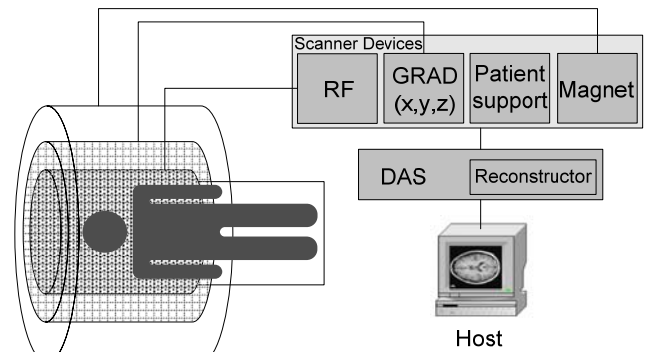


Figure 5 MRI system architecture overview

As shown in the Figure 5, the MRI system consists of the following building blocks (chains);

- *Magnet*: Provides a constant homogeneous electromagnetic field.
- *Radio frequency system (RF)*: Induces and measures a resonant field in the patient's body in combination with the main magnetic field.
- *Gradient (GRAD)*: Varies the strength of the main magnetic field at specific locations.
- *Control Data and Acquisition System (DAS)*: Acquisition and control of scan data in cooperation with the reconstructor.
- *Patient support (handling & administration)*: Patient positioning, observation, monitoring and data collection.

The documentation and archives are structured according to the building block hierarchy, resulting on abstract (top-level) archiving and detailed (low-level) archiving. Top-level archiving is about grouping functionality, whereas low-level focuses on the actual implementation. The building block method is also used as a management tool to track development. Project deliverables are expressed in building blocks and each block is subject to design, implementation and test standards. Each building block has one owner assigned who is responsible for

the contents of this block. Building blocks that require more than one area of expertise (e.g. software and hardware disciplines), have multiple owners assigned. Building block owners are listed in a database that can be accessed throughout the organization. System architects are responsible for the building block hierarchy.

Each chain acts as a viewpoint on a part of the MRI system from an abstract (documentation), implementation (hardware and software) and project management perspective (tracking the development process), where hierarchical (part-of relationships) and dependency aspects (use relationships) characterize each viewpoint (Jaring, Krikhaar et al. 2004).

Evolvability and MRI systems

Since the first Philips MRI commercial system was released on 1982, it has evolved several times leading to the present *Achieva* system. Still, the architecture of the current system has remained almost unchanged compared to the original MRI system. This architecture has proven to be evolvable enough to survive all this years, adapting changing requirements and different environments. However, some changes in the system, such as adding one degree of freedom to the patient support -which seemed to be a minor change-, has taken time and consumed tens of men years. To increase the evolvability of the system, the system should be able to adapt to those changes in a fraction of time and cost.

When dealing with such an abstract property as evolvability, it is always desirable to assess or measure it in order to know whether a system or a system design is more evolvable than another and their potential to evolve. However, as there is no formal means of doing this, the comparison is usually done through a qualitative comparison using pro/con lists and similar means. Also, the current evaluations can only be done *a-posteriori*, not at design time.

Previous works to assess system evolvability have been put to the test. The model of Rowe and Leaney, although most of this work is unpublished, based on available publications, when dealing with real systems the model seems not to be applicable. The gap between theory and practice is considerable. The method proposed by John A. Christian III provides a structured framework in which the metrics may be discussed and measured. However, as it is based on experts' opinions instead of qualities or attributes of

the real system, the reliability of the measurement is poor; different experts have different estimations of the same system. We believe that the *effort* (time, cost) necessary to adapt to changing requirements or different environments can be used to determine the evolvability of a system. Finding ways to estimate the effort necessary to evolve a system, in combination with other properties, would thus be the key to assess evolvability.

To illustrate the importance of establishing an appropriate (design-time) environment, some best practices that enhance system evolvability have been observed:

- *Identify stakeholders' (future) needs:* Some of the future requirements of the system will be based on the stakeholders needs. As the system cannot evolve in all possible ways, it is important to early identify the stakeholders of the system, and establish an effective feedback communication mechanism with them. The stakeholders determine the desired evolution path of the system. If the system is able to evolve, but it evolves in a way that doesn't meet the stakeholders' needs, it won't be able to survive. The system will be replaced with a new system -probably from competitors- (e.g. while the stakeholder is concerned about power consumption, the effort and resources to evolve the system are focused on performance, which is already good enough for the stakeholder).
- *Avoid knowledge loss:* Typically, with increasing of system complexity, the available knowledge about the system in a company decreases. Expertise and knowledge of the system are usually on the expert's minds. As time-to-market put pressure on them, they generally do not have time to transfer or share that knowledge properly. Having good documentation, structured archives, knowledge information tools, etc, is usually considered important but not urgent as it has no immediate benefit. As evolving a system requires considerable knowledge of the system, significant time is spent later searching for information. Let alone experts leaving the project.
- *Prevent legacy components:* Often, if something works, the policy of "do not touch it" prevails. As time passes by, those components become 'legacy' components;

they do essential work, however they are too expensive to replace, preventing the system to easily evolve. As they are 'old', the knowledge is probably lost and replacing them becomes hard work. In addition they may have functionalities that are no longer needed. Though as some of their functionalities are still needed the whole subsystem is maintained, consuming resources with no benefit.

- *Minimize proprietary developments:* In order to succeed in the market, each system needs to have a 'differentiator', something that makes the system different from competitors: Unique Selling Points (USPs). This differentiator, as it requires specific knowledge and proprietary developments, is typically developed in-house. The reason for keeping those subsystems as proprietary development is mainly to keep the knowledge in-house. Often, as time passes by the differentiator lacks enough support turning them into a legacy component. To prevent this, those parts that are not part of the differentiator should be moved to non-proprietary solutions.

Future work

Our next step is; firstly acquire a better understanding of system evolvability of complex systems by using the MRI as a study case. Secondly find ways to assess evolvability and to estimate the impact of change. Finally, use the results to find means to steer the design process in a direction so that evolvability is designed into the system; or how to Design for Evolvability.

5. CONCLUSIONS

Evolvability, a *system's ability to adapt to changing requirements and different environments throughout its lifespan in a time-efficient and cost-efficient way*, is a property that is vital in the industrial sector for the survival of complex systems. With evolvable systems, companies can benefit from a system that can adapt to changing requirements at a cost less than is required to build a new system. Modernization of evolvable systems is expected to take less time and reduce cost. Evolvability enables easier insertion of new technology and mitigation of the risk of obsolescence in the system. Life-cycle cost is reduced by a long-lived architecture that eases evolution rather than large-scale system redesign. In

addition, evolvability gives additional flexibility, as the customer can either reuse existing infrastructure to tackle changing requirements or develop a new product.

In this paper, a literature review regarding system evolvability has been presented. To better understand system evolvability and in order to know what is relevant for a system to be evolvable, the key drivers of system evolvability; *changing requirements, environment, effort and lifespan* have been described. Also, based on these key drivers, the evolvability definition has been reviewed and presented as a composite of other system properties.

With evolvable systems, system designers recognize that the system will have to face unpredicted situations, and tries to minimize their effect when they happen. However, nowadays there are no formal means of doing this and is usually delegated to the designer's intuition. Our objective is to find a means to steer the design process in a direction so that evolvability is designed into the system: Design for Evolvability (DfE).

To achieve the development of evolvable systems, DfE will focus on the creation of an *evolvable architecture*, the description of an *evolutionary development* process and establishing an *evolutionary environment*.

To study evolvability of complex systems we have chosen the industrial sector of high-end medical systems, specifically that of MRI scanners. Although the MRI technology is relatively young, it has evolved rapidly to adapt to new technologies and requirements.

ACKNOWLEDGMENTS

The research is conducted in cooperation with the MR group of Philips Medical Systems Netherlands. We thank all the interviewees from Philips Medical Systems for their support and contributions.

We also want to thank Gerrit Muller and Pierre van de Laar from Embedded Systems Institute (ESI) and Alexander Ulrich Douglas from Philips Research for their comments and suggestions on earlier versions of this paper.

REFERENCES

- Altenberg, L. (1994). The evolution of evolvability in genetic programming. *Advances in Genetic Programming*. K. Kinnear, MIT press, Cambridge: 47-74.

- Bensley, E., L. Fisher, et al. (1995). "Evolvable Real-Time C3 Systems." *Engineering of Complex Computer Systems*.
- Bunge, M. (1977). *Treatise On Basic Philosophy: The Furniture of the World*. Reidel, Dordrecht.
- Christian III, J. A. (2004). *A Quantitative Approach to Assessing System Evolvability*: 16.
- Christian III, J. A. and J. R. Olds (2005). *A Quantitative Methodology for Identifying Evolvable Space Systems*. 1st AIAA Space Exploration Conference January 2005, Orlando, FL. Orlando, FL., Georgia Institute of Technology.
- Cook, S., H. Ji, et al. (2000). *Software evolution and evolvability*. U. o. Reading. UK.
- Dasgupta, S. (1991). *Design theory and computer science: processes and methodology of computer systems design*, Cambridge University Press.
- De Neufville, R. (2004). "Uncertainty management for engineering system planning and design." *Engineerign Systems Symposium 2004*, MIT, Cambridge.
- Fricke, E. and A. P. Schulz (2005). *Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle*, John Wiley and Sons Ltd.
- Herald Jr., T. E., T. Verma, et al. (2007). "A Model Proposal to Forecast System Baseline Evolution due to Obsolescence through System Operation." *5th Annual Conference on Systems Engineering Research*.
- Hornak, J. (2003). "The basics of MRI." 2007, from <http://www.cis.rit.edu/htbooks/mri>.
- Isaac, D. and G. McConaughy (1994). "The Role of Architecture and Evolvability Development in Accommodating Change." *INCOSE'94*: 541-545.
- Jaring, M., R. L. Krikhaar, et al. (2004). "Representing variability in a family of MRI scanners." *Softw. Pract. Exper.* **34**(1): 69-100.
- Laar, P. v. d., S. v. Loo, et al. (2007). "The Darwin Project: Evolvability of Software-Intensive Systems." *23th IEEE International Conference on Software Maintenance (ICSM 2007)*.
- McCay, B. M. (1996). "Some Thoughts on the Quality of Computer-Based System's Architecture." *ECBS'96*: 228-234.
- Medical, P. (2007). 2007, from <http://www.philips.com/about/company/>.
- Muller, G. (2004, 2007). "CAFCR: A Multi-view Method for Embedded Systems Architecting." from <http://www.gaudisite.nl/index.html>.
- Muller, G. (2008). "How Reference Architectures support the evolution of Product Families." *CSER, Los Angeles, (To appear)*.
- Percivall, G. S. (1994). *System Architecture for Evolutionary System Development*. Proceedings of the 4th Annual Symposium of the National Council on Systems Engineering. San Jose: 571-575.
- Richards, M. G., D. E. Hastings, et al. (2007). "Design Principles for Survivable System Architecture." *Systems Conference, 2007 1st Annual IEEE*.
- Ring, J. and E. Fricke (1998). "Rapid Evolution of All Your Systems - Problem or Opportunity?" *Proceedings of IEEE 17th DASC, Seattle*.
- Rowe, D. and J. Leaney (1997). "Evaluating evolvability of computer based systems architectures - an ontological approach." *IEEE Proc ECBS*: 360-367.
- Rowe, D. and J. Leaney (1998). "Defining systems evolvability - a taxonomy for change." *Proceedings of the International Conference and Workshop: Engineering of Computer-Based Systems (ECBS '98), Jerusalem, Israel*: 45-52.
- Schulz, A. P. and E. Fricke (1999). "Incorporating flexibility, agility, robustness, and adaptability within the design of integrates systems - key to success?" *Proceedings of IEEE/AIAA 18th Digital Avionics Syst Conf, St. Louis*.
- Schulz, A. P., E. Fricke, et al. (2000). *Enabling Changes in Systems throughout the Entire Life-Cycle – Key to Success ?* *Proceedings of the 10th annual INCOSE conference Minneapolis, USA*.
- Smaling, R. M. (2005). *System Architecture Analysis and Selection Under Uncertainty*.
- Steiner, R. (1998). "Systems architecture and evolvability - definitions and perspective." *Proceedings of the 8th Annual Symposium of the International Council on System Egeining*.
- Wagner, G. P. and L. Altenberg (1996). "Complex adaptations and the evolution of evolvability." *Evolution* **50**: 967-976.