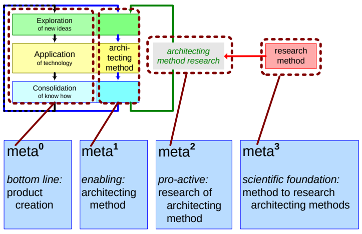
Research in Systems Architecting

-



Gerrit Muller
University of South-Eastern Norway-NISE
Hasbergsvei 36 P.O. Box 235, NO-3603 Kongsberg Norway
`gaudisite@gmail.com`

Abstract

Researching architecting methods is a very abstract activity at a large distance from actual product creation. The relation between *product creation*, *architecting*, *architecting methods* and *architecting methods research* is shown.
The technology management cycle model is explained and mapped on product creation and research. This model is used as the basis to describe an "ideal" research method that is used to study architecting methods.

version: 2.0 status: finished August 21, 2020

1 Introduction

Architecting is an extremely broad subject, taking into account many ill-defined needs, concerns, expectations, et cetera. *Architecting methods* are the result of consolidation of experience of architects. *Architecting methods* should help architects to architect. *Research of architecting methods* is again one step more abstract: it is the study and exploration of *architecting methods* in a systematic way.

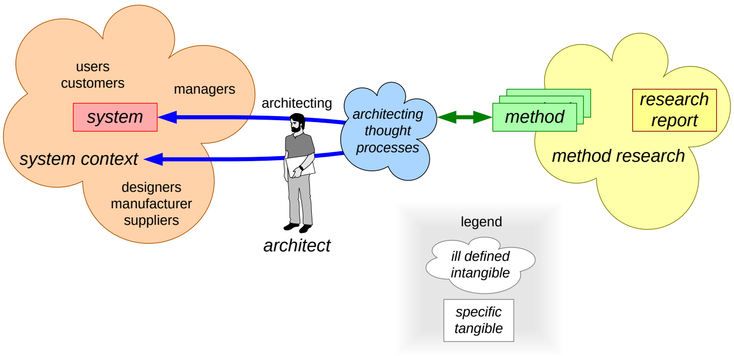


Figure 1: Research of architecting methods in the context of system design

Section 2 describes a *technology management model*. The *technology management model* is used in Section 4 to describe the research method used in this thesis to investigate an architecting method. Section 3 discusses the challenges of doing research of architecting methods in a scientific way. Section 5 describes how the distance between industrial practice and scientific research can be bridged. Section 6 describes the environment where the research takes place.

2 Technology Management Cycle

The creation of software and technology intensive products requires by definition quite some technology know-how. These technologies can be classified as *hard* and *soft*:

- *Hard* technology is the tangible engineering and scientific know-how, such as software and electronics engineering, and mathematics, physics, chemistry, and biology. The know-how from these sciences is very objective and universally applicable (the elasticity in the USA is the same as the elasticity in China). The performance of the *product* is determined by the right choice of *hard* technologies.

- *Soft* technology is the less tangible know-how of how to create a product with a team of people. Soft technologies are based on a mixture of sciences and human arts. The know-how of soft technologies is more subjective, the human factors are less well reproducible (a method working well in the USA might fail in China and vice versa). The performance of the *product creation team* depends on the right application of *soft* technologies.

The intensive use of technology in these products requires explicit management of the technology: technology management. Architecting methods are managed as part of the (soft) technology.

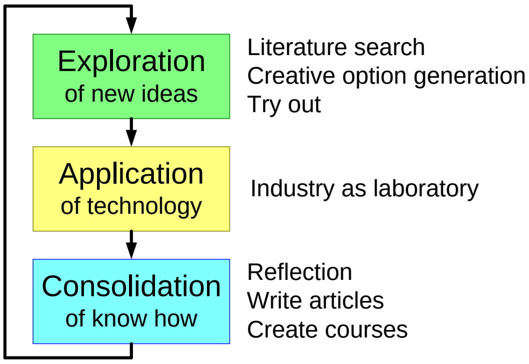


Figure 2: Technology Management Cycle

Technology management can be modeled as a cyclic process [1], as shown in Figure 2. Most of the time is spent in the application of technology, in other words in the creation of new systems. After applying the technology it is recommended to learn from this application by reflection. The learning experience can be made (partially) accessible to others by consolidating the know-how, for instance in documentation.

At the end of the consolidation insight will exist in strengths and weaknesses of the technology, both in the hard technology choices as well as in the soft technology (the approach taken). It is recommended to take this know-how as a starting point for an exploration phase.

The exploration phase should be used to refresh the designers and architects, and to open new opportunities in technology. This requires that they know the state of the art in the world, by reading literature, visiting conferences, et cetera. New technology options can be added by means of creative brainstorms . Promising technology must be explored hands-on.

In the next application phase a limited set of new technologies is applied in practice.

This thesis focuses entirely on an architecting method. The architect and the architecture are heavily involved with a lot of hard technology. However, the management of hard technologies and soft technologies other than architecting methods is outside the scope of this thesis.

3 Challenges to do Research in a Scientific Way

Science is applied in a wide range of areas, from proof-based mathematics to descriptive reasoning in human sciences, see Figure 3.

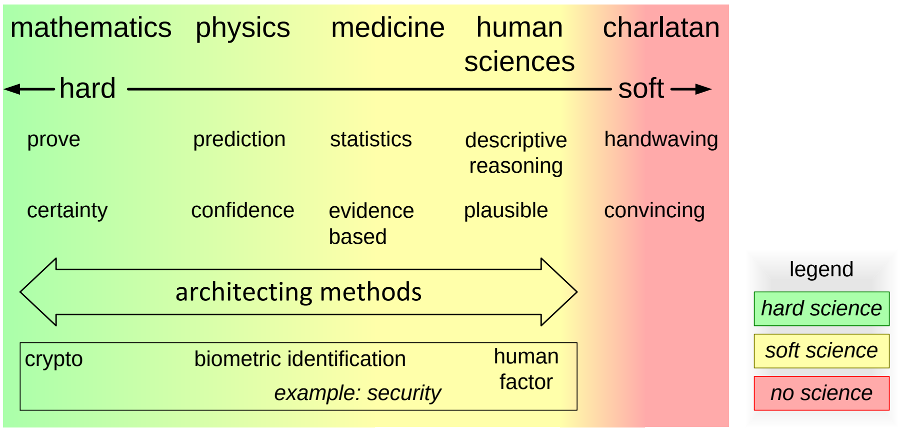


Figure 3: Spectrum of sciences

The level of certainty of the results decreases when moving from hard sciences to soft sciences. Mathematical proofs provide certainty[1], see also [2]. Physics provides a confidence level that increases by validating predicted outcomes, or it applies a *falsification* process as described by Popper [12].

Medical sciences need a lot more trial and error, where evidence is built up in extensive statistical studies. The evidence is hampered by many factors that influence the outcome of the medical study, but that are outside the control of the experimenter. Worse is that many of the factors are unknown to the experimenter and his peers. Cause and result are often more ambiguous than people realize. Despite all these disclaimers the medical sciences have created a large body of knowledge.

The human sciences (psychology, sociology, pedagogy, et cetera) have already a tremendous challenge in making statements plausible. Human behavior shows a wide variation, depending on many factors, such as culture, age, gender, and status. Individual human behavior is often poorly predictable. Case descriptions are used in a heuristic approach. The step from case descriptions to a workable hypothesis

---

[1] As far as the proof is verifiable and the verifiers can be trusted. The absolute certainty is here also decreased by the human factor: the proof is as certain as the quality of the provider of the proof and the verifiers of the proof. Automation shifts the problem to the tool, which also in some way originates in fallible human beings.

needs a lot of interpretation. Adding more case descriptions will help in making the issue more plausible, but hard evidence is nearly impossible. A more experimental approach with small scale experiments is possible, but these experiments are often highly artificial.

The scientific community dislikes the charlatans, who can be very convincing by hand-waving arguments, but in fact are selling hot air.

Architecting integrates all of these different types of sciences, from mathematical to human sciences. For instance in security design cryptographic proof is important, and also biometrics authentication. However a security solution that does not take the human behavior into account fails even before it is implemented.

Research of architecting methods is inherently the combination of hard facts in an environment full of soft factors. Most of present-day hard disciplines (mathematics, physics, electronics, mechanics, et cetera) are frightened away by the soft factors. Most of the soft disciplines (psychology, philosophy, business management) have no affinity with the complexity in the hard facts. The challenge in the systems discipline is to tackle the soft factors, with sufficient understanding of the hard side.

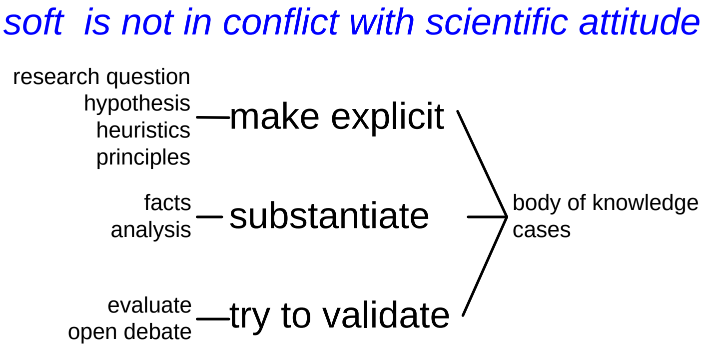


Figure 4: Soft problems can be approached with a scientific attitude

The fact that so many soft factors play a role is no excuse to stay in “trial and error” mode. The scientific attitude, see Figure 4, can also be applied to the soft kind of problems encountered in systems architecting. The Philosophy of Science has a long history. Some inspiration for the approach taken here are the *falsification* process by Popper, summarized by Tuten in [12], and the notion of *paradigms* by Kuhn, also summarized by Tuten in [13]. Popper formulated the foundation of scientific methodology, for instance based upon open discussion, testable statements and a critical attitude. The weakness of the Popper view is the notion that science progresses linearly. Kuhn introduced the notion of *paradigm shift* to show that scientific progress at some times is non linear and requires a revolution to make progress. In this thesis we want to assess the value of the architecting method for industrial application. The use of a hypothesis and evaluation criteria is less rigid than the Popper approach, but at least it supports an open debate about the merits of the method.

The first step is to make research question and hypothesis explicit. After sufficient research the heuristics and principles will become visible, which can be very powerful means to capture generic know-how, see [9] for an extensive collection of systems architecting heuristics. A nice overview is given by Pidwirny [7], using characteristics such as *neutral* and *unbiased*.

The next step is to substantiate the benefits of proposed methods with facts and analysis. The last step is to strive for validation. For many soft issues validation will be an unreachable ideal. Increasing the plausibility is then the maximum that can be achieved.

These steps together contribute to the building of a body of know-how (as all sciences do), of which a significant part will be based on case descriptions.

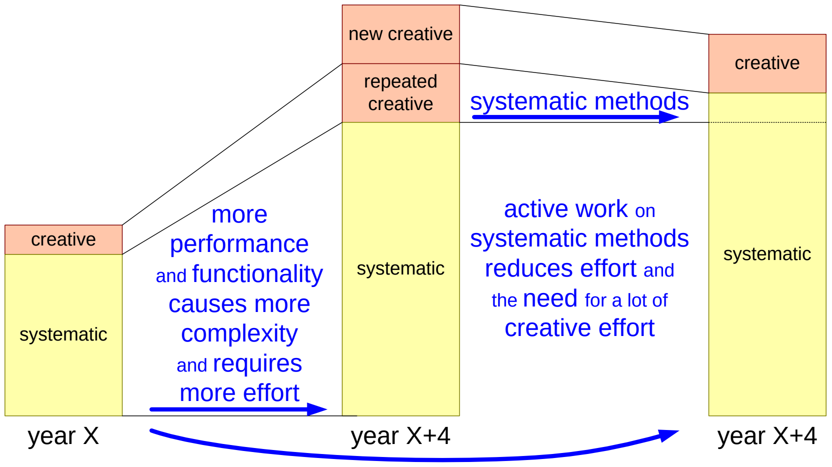


Figure 5: A scientific base is required to cope with the growing system effort. The scientific base provides a systematic approach that helps to solve known types of problems with less, more systematic, effort.

The relevance for the product creation companies is that the increasing effort of creating more powerful, but complex systems, is kept manageable. The ratio between the amount of systematic work, engineering, and the amount of creative/chaotic work should preferable stay the same. Due to the increasing complexity, in both hard and soft issues, this ratio will worsen if we are not able to make part of the system work more systematic.

Figure 5 shows the amount of systematic work and creative work. In the electronics industry the effort to create new circuits increases exponentially, more or less following Moore's Law. The phenomenon that the product needs and possibilities increase faster than our design know-how is known as the productivity gap, see for example [3]. The first bar shows the amount of systematic work at the bottom and the creative work at the top. The new development shown in the second bar, taking place several years later, in this example four years, requires about twice the amount of work. If we do not develop the system discipline a lot of the future system work will still be done in “trial and error” mode, represented by the *repeated creative* work. The new functionality, performance and complexity challenges also require *new creative* work. If the creative work of the past can be captured in more systematic approaches then the *repeated creative* work is transformed in less *systematic* work, as shown in the third bar.

One of the symptoms for this trend of increasing creative work is the relative increase of the integration period and integration effort. The lack of a systematic approach in the early design phases is solved by applying a lot of creativity in solving the problems during integration. This effect is visible in complex systems, such as MRI scanners, wafersteppers, and video processing platforms.

The message behind this figure is that product creation will always have a creative component. Providing a scientific base will never remove the need for human creativity. A scientific base will enable the effective use of the creative talent, not wasting it on problems that could have been solved in a systematic way.

Figure 5 suggests an incremental increase of creation effort. Many products, such as cardiovascular X-ray systems, wafersteppers, and televisions show such exponential growth of the effort. When developing system architecting methods the ambition should be to develop also the development of system design and implementation methods that *decrease* the desired effort. Once the know-how is captured in methods a next step in support can be made by further automation and supporting tools. Systematizing know how precedes automation and tooling.

4 Architecting Research Method

This thesis is based on research by means of the conventional hypothesis (see Section **??**) and evaluation method, complemented by case descriptions (Part III). The research starts with a research question, described in Section **??** that after some exploration work is used to formulate a hypothesis. The hypothesis is next assessed to be valid or invalid by means of criteria, see Section **??**.

The research method and the architecting methods are very abstract entities. These methods are illustrated by case descriptions. Specific case descriptions make it possible to capture the experience of the otherwise rather generic methods. The case descriptions describe parts of actual system architectures.

In the human sciences case descriptions are one of the major research methods [11]. Theory in these sciences define many abstract concepts that are difficult to make precise. Case descriptions support the definition of the concepts. At the same time, they complement the abstract concept definitions, by being very specific, thereby helping to clarify and to educate.

5 Distance between Industrial Practice and Scientific Research

The main challenge in the research of architecting methods is to bridge the distance between the pragmatic world of product creation in the industrial context and the scientifically sound research of architecting methods. Figure 6 shows the distance between the practitioners and the scientific foundation as an abstraction hierarchy.

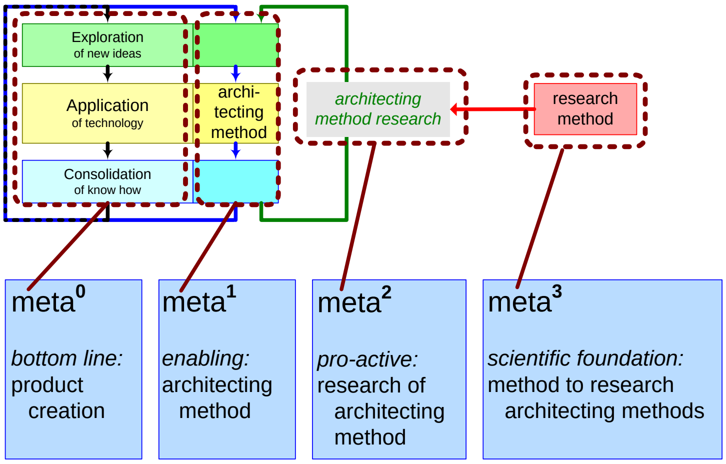


Figure 6: Moving in the *meta* direction. Research of architecting methods is two steps of indirection away from the bottom line of product creation. The scientific foundation for this work is another indirection step

The status quo in *systems* architecting is that most architects learn by trial and error[2]. These architects are directly working in the product creation process, where the bottom line is to create successful products.

The approach taken in *architecting* can be abstracted into an *architecting method*; this is the first step in the *meta*-direction. Doing systematic *research of architecting methods* is a second step in the *meta*-direction. The definition of a *research*

---

[2] A systematic foundation for *systems* architecting is lacking in the companies I have worked for. Most companies do have extensive process handbooks and quality assurance handbooks, covering documentation, verification, project management, and many more issues. However, the multidisciplinary specification and design at *system* level is left open.

I have made visits to many other companies, explicitly asking for their systems architecting approach and how they develop systems architects. I did not find any systematic foundation at *system* level in any of these companies. The companies I visited are working in the telecommunication fields, computer industry, and electronics industry.

See Chapter **??** for other work done in this area.

method (to investigate *architecting methods*) provides the systematic research with a scientific foundation: the third step in the *meta*-direction. These three levels of abstractions illustrate the different worlds of practitioners and researchers.

The drive behind this thesis is the assumption that building a scientifically founded body of knowledge will improve product creation effectiveness *directly* or *indirectly*. An example of *indirect* improvement by means of rational *design* methods is described by Parnas and Clements in "A Rational Design Process: How and Why to Fake It" [6]. The *rational design process* is in the industrial practice used *indirectly* in the later phases of the product creation for documentation and communication.

6 Research Environment

In this thesis architecting methods are studied by a retrospective analysis of a finished industrial product development. This way of working, shown as consolidation in Figure 2, makes knowledge that is obtained in the past explicit. This knowledge is consolidated to make it accessible for other people. This way of working does not work for *active* research of architecting methods, where we want to study the effects of potential method improvements.

Figure 7 shows multiple environments that can be used to study architecting methods. This thesis is based on *research by analysis* shown at the left hand side. A promising research environment is the *industry as laboratory*. Research of architecting in the limited scope of research laboratories is shown as *trial in research environment*. *Courses* and *workshops* provide an environment to obtain additional feedback on architecting methods.

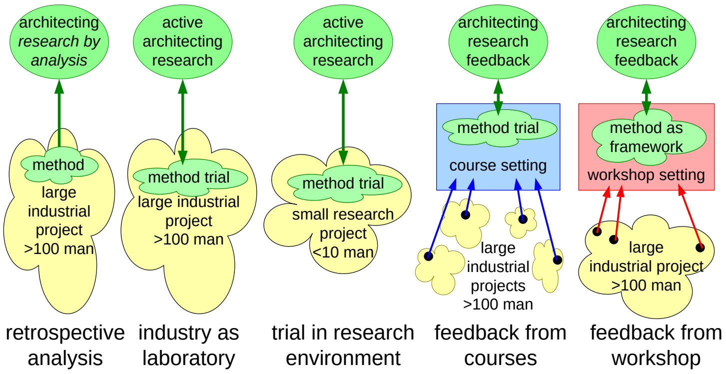


Figure 7: Obtaining practical case data of architecting methods from multiple sources

The *Industry as Laboratory* [8] approach is based on an intimate collaboration of researchers and practitioners. The Embedded Systems Institute uses this model as the basis for research [10]. The industrial environment is used to try out architecting methods. In the industrial environment the typical time and resource pressures, and the larger size (one or two orders of magnitude larger than in typical research projects) are inherently realistic for the industrial context.

Some architecting methods are also explored in *research projects*. For instance, *story telling*, as described in Chapter **??**, is used in ambient intelligence projects. The *CAFCR* model has been used many times in research workshops as a framework. Project reviews and workshop evaluations provide feedback on the architecting method for this *research project* environment.

Partial architecting methods (for instance *story telling* again) are also used in course settings, where they are applied to many different systems, ranging from silicon chips to Cardiovascular X-ray systems. More than 300 designers and architects have participated in Systems Architecting courses, using partial methods for tens of different systems. This provides valuable feedback of these methods when applied to real systems. See [4] for the course program. The entire course material, including exercises, can be found at: `http://www.gaudisite.nl/SARCH.html`.

The CAFCR method is also used within Philips as a framework for performing architecture workshops. External and internal project stakeholders present during the workshops use (parts of) the CAFCR method as a means to structure their workshop. The evaluation at the end of a workshop provides feedback for the architecting method. In Chapter **??** the evaluations from many workshops are discussed.

7 Acknowledgements "Architecting Research Method"

Henk Obbink pointed me to the source of the term *industry as laboratory*. Wim Vree challenged me to articulate the method of research and gave lots of valuable feedback during the creation.

References

[1] Hay Management Consultants. Technology management cycle. Hay Managament Consultants showed me this model in 1997/1998, taken from an article by a Japanese author. The original title of the Japanese article is unknown.

[2] Robert Hunt. The origins of proof iv: The philosophy of proof. `http://plus.maths.org/issue10/features/proof4/`, 2000.

[3] Kurt Keutzer and Richard Newton. Productivity gap. `http://lark.vmei.acad.bg/asic/lectures/intro/slide1-5.htm`, 2000.

[4] Gerrit Muller. CTT course SARCH. `http://www.gaudisite.nl/SARCHcoursePaper.pdf`, 1999.

[5] Gerrit Muller. The system architecture homepage. `http://www.gaudisite.nl/index.html`, 1999.

[6] D.L. Parnas and P.C. Clements. A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering*, SE-12., No. 2:251–257, February 1986.

[7] Michael J. Pidwirny. Fundamentals of physical geography; chapter 3a scientific method. `http://www.geog.ouc.bc.ca/physgeog/contents/3a.html`, 1996.

[8] Colin Potts. Software-engingeering research revisited. *IEEE Software*, Vol. 10, No. 5:19–28, September/October 1993.

[9] Eberhardt Rechtin and Mark W. Maier. *The Art of Systems Architecting*. CRC Press, Boca Raton, Florida, 1997.

[10] Martin Rem. Trends in embedded systems. `http://www.hightechconnections.org/presentations/martinrem.pdf`, 2004.

[11] Oliver W. Sacks. *The Man Who Mistook His Wife for a Hat: And Other Clinical Tales*. Touchstone Books, 1985. Oliver Sacks has published a rich collection of case descriptions, much more than the descriptions in this book. He explains in this book the value of case descriptions. Interesting is that although dr. Sacks is a neurologist, the case descriptions are much richer and contain many psycho social observations as well.

[12] Henk Tuten. Popper and philosophy of science. `http://huizen.daxis.nl/~henkt/popper-scientific-philosophy.html`, 1999.

[13] Henk Tuten. Thomas kuhn: definition paradigm (shift). `http://huizen.daxis.nl/~henkt/kuhn.html`, 1999.

History

Version: 2.0, date: April 5, 2004 changed by: Gerrit Muller

- many small text improvements

- legend added to figure spectrum of sceinces

Version: 1.9, date: March 17, 2004 changed by: Gerrit Muller
- small text improvement
- changed description of thge relation with the wrok of Popper and Kuhn
- changed status to finished

Version: 1.8, date: February 27, 2004 changed by: Gerrit Muller
- small text improvement
- added short description to reference
- changed status to concept

Version: 1.7, date: January 12, 2004 changed by: Gerrit Muller
- updated the presentation

Version: 1.6, date: December 2, 2003 changed by: Gerrit Muller
- moved section *Structure of this thesis* to next chapter
- many small textual changes
- changed chapter Title in *Research in systems Architecting*

Version: 1.5, date: November 28, 2003 changed by: Gerrit Muller
- renamed section ”Other sources of Practical Case Data” into ”Research environment” and significant update of this section
- many small text changes
- complete redesign of the figure showing the growing effort

Version: 1.4, date: November 5, 2003 changed by: Gerrit Muller
- added reference
- small text changes

Version: 1.3, date: September 30, 2003 changed by: Gerrit Muller
- updated text in Section ”Architecting research method”

Version: 1.2, date: September 30, 2003 changed by: Gerrit Muller
- removed redundant text in introduction

Version: 1.1, date: July 7, 2003 changed by: Gerrit Muller
- changed *review* into *workshop*

Version: 1.0, date: March 21, 2003 changed by: Gerrit Muller
- changed status to draft
- added references and citations
- rephrased to make the claims more specific

Version: 0.3, date: February 11, 2003 changed by: Gerrit Muller
- added text to explain hard and soft technology

Version: 0.2, date: February 6, 2003 changed by: Gerrit Muller
- added text about methods in the technology management cycle
- updated figure ”from Product Division to Research”
- added figure and text about moving in the meta direction
- added text about the coupling of the meta activities in time
- added unexpected observation or lateral idea to figure research of architecting methods
- added example of architecting research question, objectives et cetera
- added reference to Philips Research project management guide
- changed status in preliminary draft

Version: 0.1, date: January 24, 2003 changed by: Gerrit Muller
- Citation for industry as laboratory completed

Version: 0, date: January 10, 2003 changed by: Gerrit Muller
- Created, no changelog yet