Summary of the SISU gathering 27th September 2016

by Stein Erik Ellevseth

What happened to the companies, products and participants and why? Reminiscing and mingling SISU 20 years later.

Address of the event: Department of Informatics (Ifi) at the University of Oslo, Gaustadalleen 23B, 4th floor.
Contact person: Birger Møller-Pedersen.

At the gathering we were thirteen participants from ten companies / organizations that either participated in SISU or companies / organizations with former SISU participants. A quick presentation round of the participants:

Geir Melby, the SISU project manager, now general manager of Tellu, a spinoff company from SISU and NorARC (Ericsson Research in Norway). *Øystein Haugen,* former Norwegian Computing Center / Self employed / Sintef, now professor at HiØ in Halden. *Dag Brenna,* former Garex, now Accenture. *Dag Jarfonn* from Indra Navia (former Garex). *Øyvind Teig* from Autronica. *Rolv Bræk* from NTNU / ITEM. *Jon Ola Hove* former Taskon, now Sparebank 1. *Beint Terje Skjævesland* former Ericsson, now Commitment as. *Hans Petter Møller* former Telox, now Systemsoft. *Bernt Breivik* former Tandberg Data Storage, now Silicon Labs. *Birger Møller-Pedersen* former Norwegian Computing Center / Ericsson, now University of Oslo / Ifi. *Stein Erik Ellevseth* former Alcatel Telecom / Ericsson / Devotek / ABB AS, now Independant.

Three original SISU participating companies / organizations were present at the meeting; NTNU, Indra Navia (Garex) and Autronica. Two out of nineteen participating SISU companies and three out of twenty three organizations (incl. NTNU, NR, Sintef and Ifi) were present at the meeting. I will not speculate in the fact that the majority of the participants are no longer working in the company they worked for during SISU.

The program for the event:

13:00

- Introduction. Why and how? (Stein Erik Ellevseth).
- SISU project, a short story from start to finish (Geir Melby)
- SISU 20 years later - preliminary findings (Stein Erik Ellevseth)
- Methodology work since SISU (Rolv Bræk)
- Views from a tool supplier (Emmanuel Gaudin, PragmaDev, former Telelogic)
- Thoughts on SDL and UML, and one language for modeling and programming (Birger Møller-Pedersen)
- About education, resistance to change and unambiguous semantics (Øystein Haugen)
- The road from a SISU company to a new company (Bernt Breivik)
- What happened to the SISU companies (Stein Erik Ellevseth)

approx. 16.00 to 17.00 The road ahead (discussion)
- Automated Testing (Simula Labs)
- Object Oriented Programming and modeling of embedded systems (project)

17.30 we continued mingling and reminiscing at Kværneriet, Majorstua.

The presentations in chronological order.

Stein Erik Ellevseth, *independent*, worked for Alcatel Telecom during SISU project, gave an introduction talk. The topic was why and how?

The background for this SISU gathering is an idea I have had for a long time about what happened with the SISU results after SISU ended in 1996. The first time was in 2006, ten years later (Ten Years After) when I was on temporary leave from Kongsberg Devotek. I started with the list of SISU companies from the final report and started making contact with some of those. Then I got a new assignment and had to put this work aside.

Embedded and model-driven development has been a common thread in my career also after SISU ended. The legacy of SISU has been strong, and abstraction, quality, reduced development costs and high controllability has been important in my career. I have developed and held courses in Model Driven Development (MDD) at Semcon and Kongsberg Devotek, based on SISU / TiME methodology. I have expanded my competence to include Matlab and Matlab Simulink during my career at ABB Corporate Research. Since ABB decided to close down their research department in Norway

in 2015, I have been without permanent work. I decided to use my network where most jobs are, instead of using the time to apply for a job. I want to do work that others can enjoy and benefit from, instead of writing applications that remain in the desk drawer. Hopefully this may give me a job or an assigment later. Currently I'm looking for sponsors to write a report based on the interviews of the SISU companies I've done the past six months.
I have identified and interviewed participants and former employees of SISU companies. Based on the interviews I've done a maturity assessment based on SISU maturity model and created a table showing the maturity of the SISU companies today.

*** Geir Melby**, *general manager of Tellu*, started and managed the SISU projects and told the story of SISU from the start in 1988 until the end in 1996/97. Altogether, approximately 40 million NOK (NFR & SND support) was invested in SISU I & II. In addition, the support for SISU I pre-study of 2-3 million NOK and SISU II pre-study about 1 million NOK. (Comment from Stein Erik Ellevseth: In addition, the companies own efforts, in total well over 80 million NOK! SISU II alone had a budget of 51 million NOK, where 28 million was the companies' own efforts and 22.5 millions was support from NFR & SND.)

The SISU project was born during a dinner at SDL '88 in The Hague organized by the EBF / ITUF. Johan Lindeberg, who at the time was secretary of ITUF, said Norway needed a lighthouse project for software development. They applied for pre-project support which was used to establish the focus of languages and methods. Among others, Inger Nordgard (NFR, formerly ABB Research) and Joe Gorman (Sintef, former Elab) went along with Geir Melby on a study trip to England. There they received demonstrations of several IDEs for model driven development, including the Eclipse tool. This lead to the foundation of SISU's commitment to the language and methods to use. The early years were characterized by much discussion and disagreements for in which directions the project should go. Geir Melby told that on one occasion after a hard discussion he traveled past Lysaker where a large poster read: "We believe in Norwegian SISU", which admittedly was about Finnish tractors, but was a catchy acronym which later became “Støtte til Integrert System Utvikling[1]”. Others believed like *Dag Brenna* which frequently called it "Shit In - Shit Out[2]". SISU needed to pursue a language that expressed what we needed in embedded systems and found that the model-driven approach based on SDL was the common nerve they needed. ABB's commitment to the object orientation became leading for SISU, also based on the Mjølner project that contributed to the SDL'88 standard and from the EU project SPECS. The SISU I evaluation report made by Anders Ekeland (Formerly NR and NIFU, now SSB) from '93 was important for the start of SISU II. In a meeting with NFR , headed by Rolf Skaar at the time, Rolf Skaar said to Hauge, the head of Industry and Energy in ITUF, "why did not you tell me about this before?", whereafter roughly 40 million NOK was budgeted and designated for SISU II. This was not popular among the other groups in ITUF, who were not so happy to have their budgets reduced.

To better measure the impact of SISU project, a SISU Maturity Model was developed along the lines of CMM, the Capability Maturity Model, with five levels of maturity. Building systems are very extensive and requires more than just writing code; therefore the maturity model was developed with six disciplines; *System description, Verification & Validation, Transformation, Reuse, Product Management and Process*, each with five maturity levels.

During SISU II, more than 30 meetings with about 30-50 participants each where held. Nine SISU Forums were held, each with over 100 participants. A number of reports (about 200) were written by researchers and companies. Altogether approximately 80 deliverables were publicly available. During SISU we became a very close group with many common goals. Of the gatherings Geir remembered well were Håholmen at the Atlantic Road and sailing with Viking ships, and Lillehammer were we went on sleigh ride on Sjursjøen near by.

After SISU ended in 1996, there were several attempts to create a center for further work with the SISU methodology. It did not succeed to create a commercial center, but Ericsson chose to continue working with SISU methodology where Ivar Versvik together with Geir Melby built up a research center NorARC for languages and methods based on work from SISU. The center was a part of Ericsson's global research organization. Several key SISU scientists joined, such as Birger Møller-Pedersen, Øystein Haugen, Stein Erik Ellevseth and other Ericsson resources. Rolv Bræk also worked as an adjunct professor position at NorARC. Work on Object Oriented UML was an important result from this period.

Later around 2006 he started the company Tellu who developed an IoT platform based on a lot of the SISU methodology. Currently they are restructuring this framework from Java to Groovy, and it is based on the same principles as SISU was founded on, namely asynchronous modeling, messages and state modeling.

1 ”Supporting Integrated System Development“
2 "Skit Inn - Skit Ut"

* **Stein Erik Ellevseth**, *independent*, worked for Alcatel Telecom under SISU, presented the preliminary findings from the interviews of SISU companies.

What was SISU? SISU, Support for Integrated Systems, was a project intended to improve the profitability of Norwegian businesses through reduced development costs, better quality, short time to market and better controllability. The modeling language SDL (Z.100), SDL methodology and SDL tools was a key part of the project. One of the results from SISU is TiME, The Integrated Methodology, which is available today as an electronic book on SDL Forum's web pages - *http://sdl-forum.org/Tools/TIMe/HTML/elb/StartHere.htm*

Questions that I wanted to find answer too was "What happened and why? What is still in use? What could they imagine to do in the future?" Did the companies solve the problems that were the reason they invested in SISU methodology? I have largely figured out what happened to the companies, their products and what is in use today. I have found what happened to most participants by tracking which new companies that have moved to. I have collected a considerable amount of information about SISU technologies in use today. The participants at this gathering also participated in the introduction of SDL tools (SDT, Geode) for 20 to 25 years ago. Since then, many have abandoned the SISU methodology for different reasons. The reasons for this, I am now doing in-depth research.
.

	a) System Description	**b) Verification & Validation**	**c) Transformation**	**d) Product Management**	**e) Reuse**	**f) Process (CMM)**
1	Realization-oriented	Test-oriented	Compiled	Initial	Initial	Initial
2	partial design-oriented	Inspection	Partial generated	Version-oriented	Product- oriented	Repeatable
3	Design-oriented	Animated	Generated	Product-oriented	Domain- oriented	Defined
4	Product- oriented	Analyzed	system generated	Integrated	Integrated	Managed
5	required	Synthesized	Design transformed	Required	Global	Optimized

Tabell 1: SISU Maturity model

Stein Erik presented the SISU maturity model and the maturity of the companies 20 years later. The results are varying. Of the original fifteen SISU companies three of them no longer exists. Two ae consulting companies, therefore the first analysis will only contain nine remaining companies. It has nevertheless some bright sides with SISU technology that is still in use, such as for KDA, Indra Navia, Norsonic, Thales and Zenitel.

Of the nine SISU companies, two have increased their maturity level (KDA, Saab / Ericsson Radar) by one step the last 20 years. The others (Indra Navia, Norsonic, Thales, Petro Online) has gone back in varying degree, either because the enthusiasts no longer work in the company, do not work with SISU products (Zenitel) or that SISU methodology did not fit with which was important for the company (Assa Abloy).

Company	Bedrift	Domain	System Description			Verification & Valid			Transformations			Product Managem			Reuse			Prosess			Change '92- '96	Change '96-'16	Change '92-'16	Bedriftnavn I 2016
			1992	1996	2016	1992	1996	2016	1992	1996	2016	1992	1996	2016	1992	1996	2016	1992	1996	2016				
1	Alcatel Telecom Norway	Mil. Telecom	3	4	2	2	3	2	3	3	2	2	3	3	2	2	2	3	3	4	3	-3	0	Thales
2	Autronica	Fire Control																						Autronica
3	CAP Gemini	Concultancy																						CAP Gemini
4	Ericsson Private Network	Telecom	1	2	2	1	2	2	1	2	2	2	3	3	2	2	2	2	3	3	5	0	5	moved to Sweden
5	Ericsson Radar	Mil. Command Control	1	2	2	1	2	2	1	2	2	1	2	3	2	3	3	2	3	3	6	1	7	Saab Technologies
6	Ericsson Semafor	Mobile embedded	1	3	3	1	2	2	1	3	3	1	3	3	2	3	3	2	3	3	9	0	9	moved to Ireland?
7	Garex	Telecom	2	2	1	1	2	2	2	2	1	2	3	3	2	3	2	2	3	3	4	-3	1	Indra Navia
8	Kjell G. Knutsen	Consultancy	2	3	2	1	2	2	1	3	3	2	2	2	1	1	1	2	3	3	5	-1	4	Applica
9	Kværner Eureka	Maritime																						
10	NFT-Ericsson	Mil. Telecom	2	3	3	2	3	3	1	3	3	3	3	3	1	2	3	3	4	4	6	1	7	KDA Defence
11	NorApp	Oil downstream	2	3	1	1	2	2	1	1	1	2	2	3	1	2	1	2	3	2	4	-3	1	Petro Online
12	Norsonic	Measurement	1	2	1	1	2	2	1	2	1	1	3	3	1	3	3	1	3	3	9	-2	7	Norsonic
13	Seem Audio	Digital audio mixer	1	2	1	1	2	2	1	3	2	1	2	3	1	2	2	2	3	2	7	-2	5	Konkurs (-> Danmark)
14	Siemens	Mil. Telecom	3	4	2	2	3	2	1	3	2	2	3	3	1	1	2	3	4	2	6	-5	1	Thales
15	Stentofon	Telecom	2	3	1	3	3	2	3	3	1	2	3	3	1	2	2	2	4	3	5	-6	-1	Zenitel
16	Tandberg Data	Data storage	2	3	2				2	3	1										2	-3	-1	konkurs (-> England)
17	Taskon																							Mogul/Kantega
18	Telox	Consultancy	2	3	3	1	3	3	1	3	3	1	2	2	1	1	1	1	2	2	7	0	7	avsluttet sanntids aktiv
19	TrioVing	Hospitality	1	3	1	1	2	1	1	2	1	2	3	2	1	2	1	2	3	2	7	-7	0	Assa Abloy
	Sum																				85	-33	52	
Average			1,7	2,8	1,8	1,4	2,4	2,1	1,4	2,5	1,9	1,7	2,6	2,8	1,4	2,1	2,0	2,1	3,1	2,8				

Tabell 2: Maturity development in SISU companies

The SDL Runtime System TST (Telox SDL Tools) was developed by Elab for use in systems for G.A.Ring AS (ACD, Camex, Garex, etc.). *Geir Melby* and Telox grabbed the idea behind it after having been on a course at Elab (in 1985/86?). Telox further developed the concept to TST, and later sold the product to several Norwegian companies. ProgGen was developed by ELAB during the DaSOM project, and was funded by NFR (1987/88?). DaSOM was later commercialized by Computas department in Trondheim, but the product was closed in 1995/96. ProgGen was further developed by Sintef during SISU II and adopted by several companies also SISU.

There are several studies that shows that it is possible to reduce errors by up to 80-90% (e.g. Ericsson) in the early stages with the use of SDL and UML. Why will not the developers exploit this opportunity? Are they not smart enough, or are they just too smart? Do education fail at colleges and universities when it comes to modeling? Are developers in enterprises not clever enough to be working with models / representations at higher level of abstraction? What do students learn about modeling and programming at universities and colleges that they bring to their new workplaces?

Regarding Ericsson Private Network, Ericsson Semafor, Seem Audio, Telox and Tandberg Data, who do not exist anymore, a lot of questions arises about what could have prevented their fate as no longer being a Norwegian company. Can analysis of their fate provide some answers that may have relevance today? A cost-benefit analysis for 2001 could be interesting and doable, so that the cost-benefit analysis from 1993, which Tor Stålhane in Sintef DELAB performed, could be evaluated. What has been problematic? Some findings seems to be that the enthusiast disappeared, the (SISU) network disappeared, there were different views on abstract models, resistance to change, tool vendors were too slow to adopt the new standards, lack of hardware support (TI, Atmel, ARM, etc.) and software platforms (ISR, FSM, Scheduling, etc.) or the market changed. Open source community with "free tools", missing SDL / UML libraries, found in C libs, Java libs, etc. May be some other explanations. What is needed to achieve benefits if they compete for the same resources, i.e. programmers? *Dag Brenna* explained how they worked in Garex around 1984 and how the simple paper-based model gave great benefits, which is difficult to achieve with tools.

Preliminary results from the interviews and the maturity assessment of spring 2016 became a paper that was delivered to SAM 2016. The reviewers feedback was unanimous, which I have summarized in one sentence: "*The subject matter of this paper is very interesting and IMPORTANT where it addresses an interesting historical view with relevance for today and future consideration and provides figures on the maturity level of modeling in the industry.* "

This inspires me to continue this work ...

This analysis does not distinguish so well between what was developed during SISU and what was adopted on other products in the organization. That is why there are several questions that we want better answers to, like; If the methodology was good, why was it not used on other and new products? If it was bad, why was it not replaced with newer and better technology? It still remains to determine whether SISU methodology contributed to improved profitability, reduced development cost and increased life cycle of the products. The latter can, however, the products of Indra Navia (f. Garex), KDA Defence (f. NFT Ericsson) and Zenitel (f. Stentofon) indicate.

Finally, a sigh from a SAM 2016 reviewer:
"*UML has basically killed all other similar technologies on its way. 15 years later UML is still far from what other technologies such as SDL could do.*" Is this also a reason that maturity level has been reduced during 20 years?

* **Rolv Bræk** *from NTNU, Dept. of Telematics*, about SISU methodology, teaching at NTNU since SISU.
The essence of the methodology is still state machines, asynchronous messages, signal sequences and run time systems. The enthusiasm from SISU is no longer present in the same way. It is much more doubt and uncertainty. The question today is about what really is a showstopper for model driven development in companies. Part of the problem may appear to be action code, data, tools hassle and platform dependency.

After SISU, Rolv worked with JavaFrame and Actor Frame in cooperation with Ericsson Research (NorARC) and with the introduction of SDL methodology in AXE developments at Ericsson. Principles of Rapid and Incremental Service Development / Creation was a trend in the late '90s and which created Service Frame based on Java Frame. Collaborations, inspired by OOram, became part of the UML, which Rolv has worked extensively with (and has increasingly firm belief in).

Arctis (the forerunner of Reactive Blocks) supports the development of reusable components. The tool uses UML activity diagrams to model behavior, but can also use state machines with SDL semantics when they are better suited. In this case, the state machines generated are on the road to executable code. BitReactive, which has developed the Arctis tool, is experiencing fear of vendor lock-in from the companies they are in contact with.

The SISU textbook passed 5,000 copies recently. "Old-fashioned," said the students, who believe that the future is based on internet, web services and synchronism. Today Rolv thinks that the pendulum is beginning to swing towards more asynchronous system solutions, also in Internet- and Web development.

Some discussions about asynchronous behavior:
- Dag Brenna (Accenture) says : NAV is going in the opposite direction[3] in this respect, where the queues are being replaced with ESB and Web services. It is synchronous behavior that reigns.
- Øyvind Teig (Autronica) says : Go is the students' favorite. Go is supporting CSP, Communicating Sequential Processes, which support synchronous behavior.
- Jon Ola Hove (Sparebank 1) says : The service bus and service descriptions as integration platform is not as central as before. In comes application boxes that scale by avoiding state handling. And techniques such as SOA mode 2, micro services and DevOps are normally used today.

*** Emmanuel Gaudin** *from PragmaDev in Paris, former Telelogic,* gave us an overview (via Skype) and history from a tool supplier of SDL and UML tools. He started with the development of a LAPB protocol using the SDL tool Geode (Verilog) for Philips in 1994, with simulation and automatic code generation. He began working for Telelogic's Paris office in 1996's as FAE. Telelogic was then owned by Saab that wanted to get rid of its IT group (Combitech). Telelogic anticipated the sale and found investors that wanted to go on the stock exchange. Simultaneously they changed the direction from SDL technology (SDT or Tau G1) to UML technology (Tau G2). They failed to do it internally with Tau G2 so they purchased iLogix for its Rhapsody product that was quite well adopted. It was a chaotic time with Anders Lidbeck as CEO of Telelogic going from a quiet 80 employees technical company up to being bought out by IBM. The initial SDL tool is still today within IBM portfolio under the name IBM Rational SDL Suite. Emmanuel left Telelogic as technical director in 2000.

In 2001 he started PragmaDev to provide a tool for modeling and testing tools for reactive systems. Their tool started as a mix of UML, SDL and C / C ++, a combination of languages suitable for a specific segment. It was an SDL tool with built-in C code and they delivered their first release in 2002. In the middle of this came the end of the *.com / Telecom bubble;* and the *Middle East conflicts consequence of 9/11*. The market went down and cheap UML drawing tools began to emerge. Still PragmaDev managed to work with Nortel, Alcatel, or Airbus. At the same time they brought the SDL-RT technology to ITU-T, the SDL standardization group.

Rational had 30% of the market, Telelogic 21%. Should they have pursued Tau G2 or SDT? Later, Rational acquired Telelogic to increase their market share. From 2005/6 they could not mention SDL while visiting businesses, because it was UML that they wanted no matter what. However, it was only UML profiles that had a well-defined semantic, and it could vary from profile to profile. Most UML tools were actually only used as drawing tools. Eventually, the firms went from using dedicated UML drawing tools, to draw by means of Visio or equivalents. At this time the price of SDL / UML tools dropped from 10k Euro / USD to 1k. SysML seemed to be the only UML profile that seems to have weight enough to save UML.

Thales and Ericsson developed good internal tools (Erlang, Titan and others) which they recently outsourced to the Open Source community in the hope that these communities would be able to maintain them. With Open Source, the price dropped further from 1k to 0k! While the price went down, the quality of the tools went also down. This created dissatisfied users who eventually went back to doing it the old way, namely hand coding.

PragmaDev have been forced to deal with this and with the next release of its tool PragmaDev Studio they will offer a free version for small size systems. The strategy is that potential users first start looking at free tools. Since open source tools are quite unstable, the user will eventually look at free commercial tools and when the project gets to a substantial size they will be ready to pay for it. SDL is as old as Linux, and should really be accepted better in the industry. «UML is too much about meta modeling, something the developer is not doing."

3 "kjærringa mot strømmen"

Jon Ola Hove (Sparebank 1) commented :Taskon experienced the same fate with their tool OOram. Structured documentation was no longer in focus. It is code that is delivered. Taskon modeled in UML and OOram, but it was hard to get others to maintain these models. Why is still unclear.
Øystein Haugen (HiØ) commented : The directives from 97/98 in Ericsson promoted the use of UML on equal terms with SDL (and Telelogic´s SDT) which for a long time had a good position inside Ericsson.

*** Birger Møller-Pedersen** *from OiO / Ifi*, worked for NR during SISU, presented thoughtful points about SDL, UML and ideas of one language for modeling and programming.

SDL'92 is actually the last real SDL, i.e. with tool support, but even then they did not support all parts of the language. Some cleanup occurred in 1996. OSDL (Object oriented SDL) was attempted introduced on SDL Forum in 1987, which was made in Mjølner project where ABB was the industrial partner. SISU accounted for standardization (and hence also for the final definition, which included input from many other actors).The standardization work on OSDL started before 1988, but it was too late to get it in the 88 standard, so it came first in 1992. With SDL 2000, we got agents, mix of blocks and processes, interfaces, composites states, ASN.1 support, object oriented data, signal types, exception action. In 2010 there were only a some cleanups.

UML 1 were introduced at the time when SISU ended in 1995/96. Some years later, in 2003, SDL and MSC introduced in UML (via Ericsson in Norway and other telecom operators). Later came executable UML and SysML. Also an UML profile for SDL came around 2000, defined in Recommendation Z.109 SDL with UML.

Modeling must not necessarily be graphic (comment from *Geir Melby*). Birger believes that in the future we see envisaged one language for both modeling and programming (such as Scala, Groovy and UMPLE). "Umple" is an initiative for Model-Oriented Programming.

Some skepticism has built up around MDE, because developers have discovered major deficiencies. They now favors agile development. There are not many good examples of enterprises that have achieved good results by investing in MDE (SDL and UML). There are however many anecdotes of poor results. Most companies are concerned about maintenance of existing systems, and most MDE methods and tools do not support this procedure well enough. Developers (and students) favors new programming language rather than modeling. It seems that it is more important to be a good programmer than to model. Graphic modeling is perceived as providing poor productivity because modeling languages and tools are not good enough. They are not suited good enough for team development. It is code that sells, and debugging and testing happens at code level.

It is only 50 years ago that Simula was developed at Norwegian Computing Centre. This will be celebrated next year.

*** Øystein Haugen** *from HiØ*, worked partly for Norwegian Computing Center and as an independent during SISU, talked about lecturing, resistance to change, and that the world did not take the direction we anticipated about unambiguous semantics 20 years ago. Traces after SISU are not so easy to find today 20 years later. Participants learned a lot which they have brought forward in their careers. Later there were more problems after the *dot.com* rise and fall, and *telecom's* fall with the expensive UMTS licenses. The licenses required that the telecom providers had to change its strategies based on the decline in revenues that were based on sales of GSM products.

After SISU ended he tried to continue to use the SISU methodology with SDL in Ericsson, where they met tough resistance. Powerful forces advocated the use of UML rather than SDL. These forces had contacts in the right "circles" in the organization. It was not necessarily the best technology (in their opinion) that was decisive, but being in the right place at the right time when decisions were made. Rational with *Ivar Jacobsen*, *Bran Selic* and *Jim Rumbaugh* gained ground with the use of UML. *Birger Møller-Pedersen* and *Øystein* therefore chose to devote themselves at influencing UML. Unlike the standardization of SDL-92, there were many (especially Americans) who did not see the benefits of an unambiguous semantics. It should be the user to decide! In system maintenance there is only one description that will always eventually remain. With non-executing models, the main description will always be the source code and therefore the models crumbles.

In Ericsson (with *Birger Møller-Pedersen*), they joined forces with others from real-time environment, especially from telecom (Motorola *Thomas Weigert*, Alcatel, Nokia and *Bran Selic* from Objectime). They wanted to make UML better, which to a certain extent they achieved with UML 2 around 2003. Today several attendees from that time look back and regret that UML 2 did not became more precise. Eventually the initiatives was going in the right direction with composite structure, a UML action language and executable UML. Some have said that the best solution for an action language for executable UML, is SDL (with its UML profile). In these days (2016) the first version of a precise semantics for UML state machines is created.

At IFI Øystein later used IBM Rational Software Modeler (RSM) in teaching, but the tool was too heavy and certainly not flawless. He used his students to create a series of plugins that improved RSM and defined a UML 2 profile that was executable with support tool for debugging and analysis, as well as an improved editor for sequence diagrams. Lately, ThingML being developed at Sintef, a textual modeling language where state machines are the key concepts of behavior, which Øystein now is using in teaching at Østfold University College (HiØ).

Lancaster University made an empirical report telling about MDE used in businesses and it turns out that Domain Specific Languages, DSLs, is more popular than one might expect, however proprietary. Further that they are afraid of vendor lock-in. UML profiles are in use. UML is otherwise in use everywhere (also at HiØ), teaching modeling using UML. UML is commonly used as "sketching" rather than a formal definition.

Variability Modeling is a DSL and a language for managing product lines (based on the projects Families, MoSiS and VARIES).

Verification & Validation was important in SISU and should be at least as important today. Automation of testing is very popular. Very many companies are today Test Oriented (accoring to the SISU maturity model) with contributions of Inspections. Animation and Simulation are used to some extent. Analyzed and Synthesized does not exist in some SISU business, nor in limited extent in companies that the SISU companies can be compared with.

Did we promise too much 20 years ago? *Øystein* presented the AI history of *Bran Selic*, which promised much, but disappointed due to lack of computing power and limitations. After the "AI Winter" which caused AI to become an inflamed word, and therefore choking the cash flow. Later they called it expert systems and today it is called Machine Learning. Is Model Driven Engineering (MDE) a similar story?

*** Bernt Breivik** *from Silicon Labs*, worked for Tandberg Data Storage during SISU, told us about bringing SISU methodology to a new business. He believes that good economics is important in order to allow improvements. Technical success is often based on economic success, not vice versa as we expected from the SISU goals. We developed a "tribal languages" in SISU, that many of us still speak. In TDS they were allowed to develop their methods because the economy was very good at that time. They were market leader in data storage and had the largest suppliers in its customer base, including IBM. Several tools suppliers held presentations, among them with SA / SD tool. TDS joined SISU in 1990, and a dozen developers attended a two week EEU course in Trondheim. Later he used SISU methodology in the development of a target laser designator at Vinghøg, with state modeling, asynchronous message passing, runtime system system, etc. Today he uses little or nothing of SISU methodology.

*** Stein Erik Ellevseth** about the SISU companies.
Based on the interviews, I've created a table for each company with their maturity per 1992, 1996 and 2016. Comments on the criteria for each maturity assessment is placed in the table per discipline. Below I have made a short summary of SISU work in the companies, during, after and 20 years after SISU:

Alcatel Telecom Defense (now *Thales*). Implemented activities in SISU II that was largely successful, achieving improvements in many areas, especially in the beginning. After SISU ended, improvement of methods and tools (also from SISU) competed with organizational changes and process-related activities. This resulted in varying commitment from management. Today UML to some extent is in use at a higher level, and some SDL on existing products. No new product uses SISU methodology.

Ericsson Private Network (*closed 1997*). SISU was a catalyst for improvement work in the enterprise. After SISU ended, this activity was terminated in 1997 when the product was transferred to Sweden and the department was later closed down.

Ericsson Radar (now *Saab Technologies Norway*). Participated in SISU through the EU project REBOOT. They withdrew from SISU in 1993 due to high work pressure and that the other companies in SISU who also worked with reuse was so different that it was difficult to cooperate. Today it is unclear how much of SISU methodology is visible in the company.

Ericsson Semafor (*closed 2013*). SISU was important for improvement work in the enterprise. An important contribution was that they could show that the SISU methodology and tools was suitable for small complex (embedded) micro-controllers with high real-time requirements and limited space requirements. Their SISU activity ended in 1995 when they had achieved their objectives. In 1997 Ericsson Semafor became a part of the Ericsson Communications System, later EMP - Ericsson Mobile Platforms. They developed mobile phones and platforms for mobile communication devices where the development environment was based on traditional coding in C / C ++ and incremental development based on the waterfall method, later scrum. Development environment based on SISU

methodology and SDL were eventually discontinued as Ericsson Semafor products were phased out. The department's products was in 2009 transferred to ST-Ericsson, which was a merger of ST Electronics and Ericsson Mobile Platforms. ST-Ericsson was split up again in 2013 and 1800 modem engineers were taken back to Ericsson.

Garex (now *Indra Navia*). The quality of the delivered systems were significantly improved by participation in SISU in that they adopted SISU methodology in business. After SISU they continued to use SISU / SDL methodology and tools on existing Garex products. They merged in 1995/96 with Normarc and took the name Seatex Garex. After the merge Normarc products were not subject to the use of SISU / SDL technology. Today OOram is no longer in use, and the code is maintained manually. TST is still used on previous OOram based products. SDT tool chain used in a new product.

NFT Ericsson (now *Kongsberg Defense*). When they look at the results, the SISU project was characterized as one of the most useful projects NFT-Ericsson has participated in. After SISU they continued to use SISU / SDL methodology and tools. Eventually they replaced CHILL code skeletons with C skeletons and used the same SDL on the new hardware. Today, SDT, ProgGen and TST are still in use.

NorApp (now *Petro Online*). SISU contributed to better awareness regarding process and quality assurance, which was used more methodically. After SISU they improved their profitability although they made losses during the first years. Now the economy is very good. Today the SISU methodology is no longer visible in the company.

Norsonic. Besides the academic results obtained, the SISU project has been very helpful for the business by being able to exchange experiences and gain inspiration from other similar environments. After SISU, SDL and code generation was gradually less used. Today state orientation and TST is still in use in their products.

Seem Audio (later *Telecast, and closed 2003*). SISU was a great inspiration for improvement, both for Seem Audio and following companies. Particularly helpful was the sharing of experiences with other companies. SISU's standardization work gave smaller companies an opportunity to indirectly influence tool vendors. After SISU came a turbulent time with technical problems, delays and several bankruptcies.

Siemens Defence (now *Thales*). The main objectives of SISU II was to improve the process of managing, analyzing and tracking customer requirements in the development process. The company had great benefits from using the SISU's textbook and SDL courses. SISU was a useful forum to exchange information, learn more about new opportunities with SDL and MSC, and not at least getting constructive help to improvements from SISU consultants. After SISU the methodology was not easily transferable to other areas, such as KKIS.

Stentofon (now *Zenitel*). The company achieved good results in SISU which inspired to improvement in the 6 disciplines of SISU II. In addition, they observed that the development of product Alphacom with better methodology and tool support led to a good and stable product with few flaws. After SISU ended there were no new products from the merger with Zenitel that was subject of SISU methodology. Today state orientation mindset exist, but SISU / SDL is only used on the existing product Alphacom with tool chain based on SDT, ProgGen and TST.

Tandberg Data Storage (*closed 2009*). The company achieved its main goals of the SISU project and its planned improvements, which was 30% reduction in development time and that half of the developers was to have good SDL and SISU methodology knowledge. After SISU ended, different views on abstract models appeared in the development of their data storage devices. SDL was eventually abandoned and they rewrote the whole system to ISR based execution, round-robin scheduling and hand coding. Customer (IBM) requirements greatly influenced the development of quality, but not in areas where SISU methodology focused. After unsuccessful efforts on a new storage standard sales went down which caused cutbacks, and the company shut down in 2009.

Telox (*closed 2000*). The aim was to build up expertise in formal use of SDL-92 and build a complete tool chain for software development, and business support and maintain results from SISU II. After SISU ended the employees who worked with real-time development eventually left the company and activity was discontinued around 2000.

TrioVing (*now Assa Abloy*). The company had benefited from participation in SISU, especially in the development of a common quality system. After SISU ended, their administrative computer systems became more important than the reactive aspects of lock control and access control. Today the SISU methodology is not visible in the company.

Among the former SISU companies the SDL tools SDT (Rational SDL Suite), TST (Telox SDL Tools), and ProgGen are still in use in KDA Defense (former NFT Ericsson), Indra Navia (former Garex), Zenitel (former Stentofon) Norsonic and Thales (former Alcatel Telecom).

After meeting note:
It was a memorable and positive meeting. Many sensed something of the atmosphere from the SISU time in what was presented and debated. All participants were very pleased with this gathering. I look forward to the road ahead whatever it may be. It's too early to say anything specific about the way forward, but I'm optimistic. The meeting was about giving a historical view on model-driven development, as we did it in SISU. When we eventually find out what we can learn from it, I think that something concrete can come out of this initiative.

The atmosphere on this gathering motivates me to continue the work to find out what happened to the companies, products and people. And not least, how it happened and why?

We did not have time to discuss a project proposals for Automated Testing from Simula Labs or Object Oriented modeling and programming of embedded systems. The latter Birger Møller-Pedersen brought up to some extent in his presentation. I hope to come back to them at a later occasion.

Finally, thanks to Birger from UiO / Ifi who provided meeting room and an honorable mention to Øyvind Teig from Autronica of Trondheim who went to Oslo in one errand to attend the gathering, and at his own expense.

mvh,
Stein Erik

Epilogue. Language and tool support are essential. Later in the evening we talked about the formal tool initiatives daSOM and OOram, both of which were very promising and got very far. Sadly daSOM lost in combating with SDL tools from Telelogic / Verilog and OOram lost to UML tools from Rational. What lessons can be learned from this that BitReactive's toolset Reactive Blocks can benefit from? Or other similar good initiatives such as ThingML, etc. What do they need to succeed? Do Reactive Blocks meet necessary criteria to become a tool that supports modeling as well as programming (embedded) systems? Today Reactive Blocks only support one action language, but in order to reach a larger (embedded) market must surely C / C ++, C # and maybe BETA be supported.