2 Introduction to the Method

About TIMe - The Integrated Method . . . 2
How to learn TIMe . . . 5
Themes . . . 6
TIMe Electronic Book . . . 7
The SISU project . . . 8
Results . . . 8
Partners . . . 9
The Authors . . . 10
Rolv Bræk; SINTEF Telecom and Informatics . . . 10
Joe Gorman; SINTEF Telecom and Informatics . . . 11
Øystein Haugen; Ericsson Norway . . . 11
Birger Møller-Pedersen; Ericsson Norway . . . 11
Geir Melby, Ericsson Norway . . . 12
Richard Sanders; SINTEF Telecom and Informatics . . . 12
Tor Stålhane; SINTEF Telecom and Informatics . . . 12
About TIMe . . . 14
Copyright SINTEF 1997, 1998, 1999 . . . 14
About this version . . . 14
Future plans . . . 15
Revision table . . . 15
About SINTEF . . . 17
List of figures . . . 18
List of definitions . . . 19

Introduction to the Method

About TIMe - The Integrated Method

The Integrated Method (TIMe) is not just a method, but a Methodology, meaning a collection of methods and languages with guidelines for when to use them. We call it The Integrated Method only to keep the name short.

TIMe is a continued improvement of the methodology that was brought forth by The SISU project (p.2-8). TIMe seeks to achieve the higher process maturity levels. TIMe supports Design oriented development, and even goes one step towards making the vision of Property oriented development come true.

TIMe recommends a combination of languages and notations that may be used to analyze, model and ultimately build systems. These are:

- UML - The Unified Modeling Language [146],
- SDL - Specification and Description Language [102], [108] and
- MSC - Message Sequence Charts [110].

In this respect, TIMe integrates these languages and notations. But the foundation of TIMe is independent of these languages, which means that the future of TIMe is not highly dependent on the future of these languages.

TIMe provides:

- a set of system development activities that covers most of the system development process, with emphasis on the activities leading up to implementation,
- guidelines on object and property modeling in general, and particularly how to do it in UML / SDL and MSC respectively, and
- tutorials in UML, SDL and MSC.

Object oriented

TIMe is truly object oriented in its approach. It defines its own underlying object and property models, and contains detailed guidelines on:

- how to make analysis object models using Unified Modeling Language (UML),
- how to make design object models using Specification and Description Language (SDL), and
- how to make interaction property models and Use Cases using Message Sequence Charts (MSC).

TIMe is characterised by:

Abstract models

- Emphasis on *abstract* models and descriptions: Abstract descriptions leave out implementation specific details and let the developers focus on functionality.

Property models

- Focus on (external) *properties*: Objects are the building material from which systems and components are constructed. Property descriptions are used at an early stage of development to express the properties *required* from a system or an object. At a later stage they are used to express the properties actually *provided* by a system or component.

Service orientation

- Users tend to think in terms of services and interfaces. Therefore TIMe recommends use of separate property models for services and interfaces. These models are used for high level service engineering, and for sythesising object designs that provide the services.

Roles

- Strong *object-property relationships*: Roles are used to describe properties, and are related to object designs by projection. Roles are used to link properties and objects. Projections are used for synthesis of new objects and for documenting existing objects.

Design for reuse

- Planned *variability* and *reuse*: TIMe seeks to make generic system *families* which may be adapted as easily and safely as possible to the needs of particular systems. Components for reuse across families come from general domain descriptions. TIMe describes a cost-effective way to define instantiation of particular systems by defining the general parts by reference to the family description, detailing only what is special for that particular occurrence, i.e. its *configuration*.

Synthesis

- Design *synthesis*: Property oriented design involves:
  - Decomposing required service and interface properties into object properties.
  - Synthesizing object designs from required object properties, by transformation and by composition, taking reuse into account.
  - Comparing properties: required against provided (validation).

Design with reuse

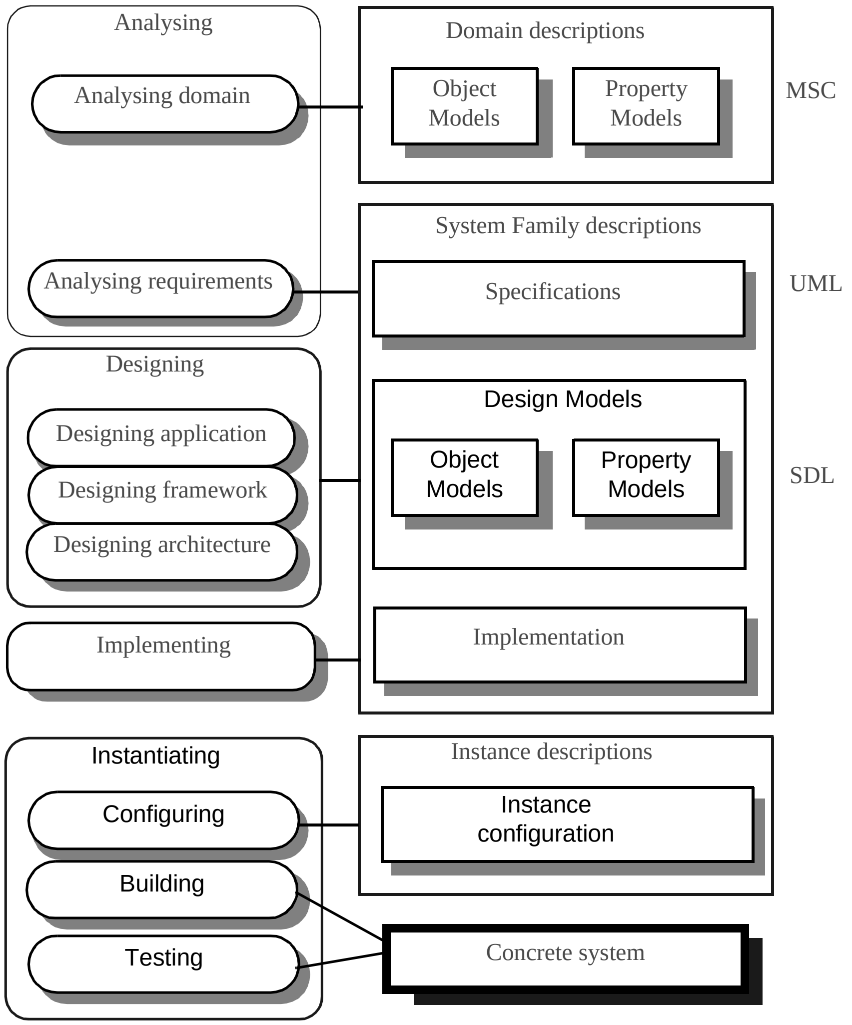
  - Searching for components with provided properties corresponding to some required properties.
  - Composing properties corresponding to object composition.

TIMe is centered around a set of models and descriptions capable of expressing domain knowledge, specifications in terms of external properties, system designs in terms of structure and behaviour, implementation mappings and system instantiation

Like most other similar methods, TIMe distinguishes between Analysis, Design, Implementation and Instantiation (see Figure 2-1 (p.2-4)).

Figure 2-1: TIMe activities, descriptions and languages

Open figure



The distinction between Domain and System Design is not particular for TIMe. What is special, however, is that:

- design is split between
  - *application design*, where the functionality of the system is design,
  - *architecture design*, where the non-functional properties are taken care of, and
  - *framework design*, that defines types of systems with the same infrastructure (e.g. supporting distribution) where the application specific parts are singled out to be redefinable in specific systems.
- the complementary *object models* and *property models* are used both for domain and system analysis, and for design.

How to learn TIMe

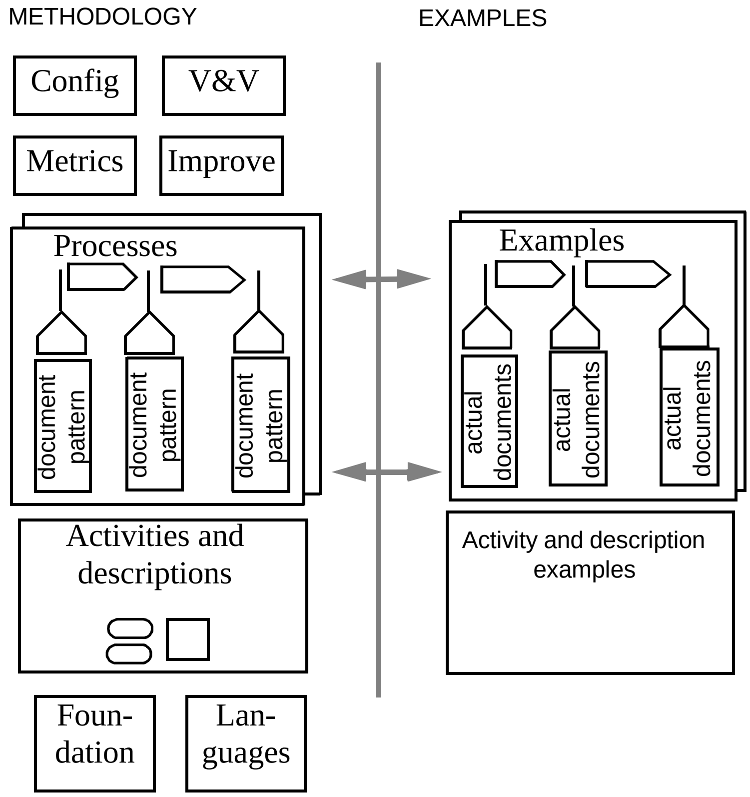
There are several ways to learn The Integrated Method:

- By reading. Enter the TIMe at a glance theme; read it on-line or print it out and read it on paper.
- By example. Enter the Examples theme and see how the methodology works on a practical example. A complete worked example of descriptions and documents as they appear at various stages of development is presented there.
- By process. Enter the Process models theme, and see how activities may be invoked and descriptions evolved.
- By activity. Enter the Activities and Descriptions theme and learn about the activities and descriptions, and their rules and guidelines.

These ways complement each other and are systematically linked.

Figure 2-2: Methodology overview

Open figure



Themes

The main themes of TIMe are:

- Introduction to the Method, which is the theme you are reading just now.
- Introduction to the Book, which tells you how to use the media.
- TIMe at a glance, which gives a first introduction to the Methodology. We recommend that this be read next.
- Foundation of TIMe, which deals with underlying principles of TIMe.
- Activities and Descriptions, which details all the main activities that are performed when developing systems using the Integrated Methodology, and which descriptions should be produced at the different levels of abstraction. Practical guide lines and strategies are provided all the way from domain analysis to system implementation.
- Process models, which presents the “project management” view, showing how the various activities are performed and how the descriptions evolve in the context of real projects.
- Examples. A comprehensive example is used to illustrate the various descriptions and how they are developed and evolve.
- Object and Property models presents the languages used for expressing object models and property models, and how to align the two.
- Languages and Notations provides language tutorials for UML, SDL and MSC.
- Verification and Validation deals with the corrective quality assurance activities.
- Process Improvement deals with the introduction of TIMe into a company and later process monitoring and improvement.
- Metrics deals with the measurement of the development process.
- Configuration Management deals with the handling of evolving software components

Read the Introduction to the Book to learn how to use the media, or TIMe at a glance for a first impression of the Methodology.

TIMe Electronic Book

It is our experience that a methodology book is used in several ways that are somewhat conflicting:

- as a textbook to be read from the start to the end;
- as a quick reference to language issues;
- as a source of examples and design solutions;
- as a source of templates for specific descriptions;
- as a guide to process improvement;
- as a concise methodology handbook.

Using an electronic medium using hyperlinks we hope that all these needs may be satisfied.

Read the theme Introduction to the Book to learn more about how to use the material on-line, and how to print any or all parts of the book.

The SISU project

TIMe was initially conceived in the SISU project, which was a Norwegian technology transfer program aiming to improve the productivity and the quality of Norwegian companies that develop systems within the real-time domain.

The project started in 1988 and ended in 1996, and involved some 20 Partners (p.2-9) from industry (including Alcatel, Ericsson and Siemens) and research institutes, including the Norwegian Computing Center (that natured Simula in the '70's) and SINTEF.

The project has contributed to higher profitability in the participating companies by increasing the their capability to achieve:

- right system quality
- shorter time to marked
- reduced development costs
- improved project control

To achieve these goals the project focused on:

- System descriptions
- Verification and Validation
- Transformations
- Configuration and reuse
- Process improvement

Results

Object oriented SDL (SDL-92)

The project has been active in the development of the object oriented extensions to SDL that became part of the language in the 1992 version of Z.100 (popularly called SDL-92) [102]. It also contributed to the Methodology Guidelines for SDL (appendix I to Z.100) [104].

MSC-96

The project participated in the MSC development within ITU, including the structural concepts that became part the 1996 version of MSC (popularly called MSC-96) [110].

Textbook

SISU developed a forerunner of TIMe, the SISU methodology, which is described in the textbook by Rolv Bræk and Øystein Haugen: *Engineering Real Time Systems. An Object-Oriented Methodology using SDL* [24]. Presently, some 600 engineers have been through a training course based on this book, and some 15-20 Norwegian companies use the SISU methodology actively in their product development. Most of them have tool support for SDL with code generation.

Web site

For more information about the results of the project, and to download public reports, visit the SISU web site at http://www.sintef.no/sisu

Partners

The following companies and research institutes have been active participant of the SISU project (SISU-I and SISU-II):

- Alcatel Telecom Norway,
- Autronica
- CAP Gemini (Computas),
- Ericsson Private Networks, Norway,
- Ericsson Radar, Norway,
- Ericsson Semafor, Norway,
- Kjell G. Knutsen,
- NFT-Ericsson,
- Norapp,
- Norsonic,
- Scanpower,
- Seatex Garex (Garex)
- Seem Audio,
- Siemens (Norway),
- Stentofon,
- Trioving,
- Tandberg Data Storage,
- Telox,
- SINTEF,
- Norwegian Computer Center (NR).

The Authors

The persons listed below have directly contributed to this book:

- Rolv Bræk; SINTEF Telecom and Informatics (p.2-10)
- Joe Gorman; SINTEF Telecom and Informatics (p.2-11)
- Øystein Haugen; Ericsson Norway (p.2-11)
- Birger Møller-Pedersen; Ericsson Norway (p.2-11)
- Geir Melby, Ericsson Norway (p.2-12)
- Richard Sanders; SINTEF Telecom and Informatics (p.2-12)
- Tor Stålhane; SINTEF Telecom and Informatics (p.2-12)

In addition a large number of people have commented on its contents, and helped clarify our ideas. These are colleagues of ours, participants of The SISU project (p.2-8), and international professionals. We thank you all!

We particularly wish to thank the people at Ericsson who have helped pilot the methodology, that worked on the HTML version, and that provided valuable feedback: Tim Papas, Steinar Lundeberg, Stein Erik Ellevseth, Pål Berg and many others.

On this version, Therese Nilsen and Per Holager of SINTEF Telecom and Informatics have helped compile the final version for publication.

Rolv Bræk; SINTEF Telecom and Informatics

Rolv graduated from the University of Trondheim, Department of Electrical Engineering in 1969 and has been working with SINTEF since 1973, where he is Principal Research Scientist. He is also Professor at the Norwegian University of Science and Technology in Trondheim (NTNU).

Rolv has been working with software engineering methodology since the early 70’s in close cooperation with Norwegian industry. One of the most reputed projects was the MAREIK project in the INMARSAT system 1979-81.

Rolv has been working with formal description techniques for decades, and fathered the SOM[1] language and methodology, a parallel development to SDL.

Rolv participated in The SISU project (p.2-8), with responsibility for Methodology. He is currently occupied with introducing TIMe to Ericsson, and is also contributing to the Z.109 standard for SDL with UML.

Rolv is co-author of *Engineering Real Time Systems - An object-oriented methodology using SDL* [24].

1. SOM initially stood for Structure-Oriented Modeling, and was later changed to SDL-Oriented Methodology. SOM is no longer supported by SINTEF, but TIMe contains many of the basic principles of SOM.

Joe Gorman; SINTEF Telecom and Informatics

Joe studied Computer Science at the University of Glasgow, where he gained his Honours Degree in 1977. After working in Scottish Universities, he started work at SINTEF in 1986.

Joe is involved with contract research work with Norwegian industry, and in international co-operative research funded by the European Commission. His main research interests are software engineering, software development methodologies, compiler techniques and configuration management.

In The SISU project (p.2-8) Joe was responsible for Configuration Management.

Øystein Haugen; Ericsson Norway

Øystein graduated from the University of Oslo in 1980, where he was assistant to Kristen Nygård for a period.

Øystein worked at the Norwegian Computing Center for 4 years as project leader for a Simula machine. From 1984-88 he worked at SimTech, and from 1988-1990 as senior research scientist at ABB Corporate Research, Norway. Øystein was an independent consultant while working on his Ph.D., until he joined Ericsson Norway in 1997 to work on software methods and languages. Øystein is Associate Professor at the University of Oslo.

Øystein participates in the standardization work in ITU, first in the development of the object-oriented extension of SDL that where incorporated in the 1992 version of SDL (“SDL-92”) [102], and later as Associate Rapporteur for MSC in ITU-T Study Group 10, that produced the 1996 version of Message Sequence Charts [110]. He is currently Rapporteur for MSC in ITU-T, responsible for the development of MSC standard year 2000.

Øystein is co-author of *Engineering Real Time Systems - An object-oriented methodology using SDL* [24]. In The SISU project (p.2-8) Øystein had responsibility for Verification and Validation activities.

Birger Møller-Pedersen; Ericsson Norway

Birger graduated from the University of Copenhagen in Denmark in 1976.

Birger worked at the Norwegian Computing Center from 1977 to 1996, where he specialized in object oriented languages, including designing and implementing compilers for Simula, designing Beta, and later adding object orientation to SDL.

Birger joined Telenor Research and Development (the Norwegian PTT) in 1996, where he amongst other things worked on Java in TMN.

In 1998 Birger joined Øystein and Geir at Ericsson NorARC, where he has continued his work with TIMe and SDL. Birger is also Associate Professor at the University of Oslo.

Birger participates in the standardization work in ITU, first as Associate Rapporteur in the development of the object-oriented extension of SDL that were incorporated in the 1992 version of SDL (“SDL-92”) [102]. He is currently participating in the ITU-T Study Group 10, working on the next version of the SDL standard scheduled for the year 2000. Birger is Associate Rapporteur for the harmonization of SDL and UML, in the coming Z.109 standard.

Birger is co-author of *The BETA Programming Language* [123] and *Systems Engineering Using SDL-92* [144].

Geir Melby, Ericsson Norway

Geir worked at Telox until 1988, where he acted as manager and consultant. Geir participated in the development of embedded software systems for industrial companies and in Telox’ own products, that included a run-time support system for SDL (Telox SDL Tools).

Geir left Telox to lead The SISU project (p.2-8) from 1988 to 1996.

Geir joined Ericsson Norway in 1996, where he is manager of the Software Engineering Laboratory of the Norwegian Applied Research Department (Ericsson NorARC).

Richard Sanders; SINTEF Telecom and Informatics

Richard graduated from the University of Trondheim, Institute of Informatics in 1984, Mastering in Computer Science and Telematics.

He worked for 3 years as a consultant with Computas (now a part of CAP Gemini), developing embedded software for communication systems using SDL/SOM, and working on a CASE tool for SDL (DASOM).

Richard worked at Stentofon (now Stento) from 1987 to 1994 as designer and later software manager, developing a new generation communication exchange, where automatic code generation from SDL was introduced in 1988, to our knowledge the first industrial project to do so. Stentofon where participants of The SISU project (p.2-8) from the start.

Richard joined SINTEF in 1994, where he has been working with development methodology (in The SISU project (p.2-8) and in Mechatronics), and as a UML/MSC/SDL designer in industry projects. He is currently occupied with introducing TIMe to Ericsson Norway. Richard also lectures at the University of Science and Technology in Trondheim (NTNU).

Tor Stålhane; SINTEF Telecom and Informatics

Tor graduated from the University of Trondheim, Department of Electrical Engineering in 1969. He holds a Ph.D. in statistics from the University of Trondheim, Department of Mathematical Sciences from 1988.

Tor has worked at SINTEF since 1969, specializing in Software Reliability and Process Improvement. Since 1997 he is also a professor at the Stavanger Polytechnic, where he teaches quality assurance and process improvement in software development.

In The SISU project (p.2-8) Tor was responsible for Software Metrics.

About TIMe

Copyright SINTEF 1997, 1998, 1999

For information about commercial terms or for comments on TIMe, please contact:

TIMe
SINTEF Telecom and Informatics
N-7465 Trondheim
Norway

Tel: +47 73 59 30 00
Fax: +47 73 53 25 86
e-mail: time@sintef.no
WWW: http://www.sintef.no/time

Full text of the ITU standards ([102] through [110]) can be obtained from the ITU Sales Section, Place des Nations, CH-1211 Geneva 20, Switzerland (see http://www.itu.int/publications or Tel. +41 22 730 51 11, Fax +41 22 731 51 94, or email sales@itu.int).

About this version

This version 4.0 of the TIMe methodology is the second commercially available version. The major improvements compared to version 3.1 are:

- UML replaces OMT and OMT+-: all examples have been revised, and a Tutorial on UML has been included
- The OMT+- Tutorial has been discontinued
- TIMe HTML version for the parts that are best suited for on-line use
- Improved Activities and Descriptions theme
- Dictionary added

Some of the previous TIMe versions (p.2-15) where only available to participants of The SISU project (p.2-8) or to Ericsson.

We have put a lot of effort into this book, but we have not reached what we consider to be a finished product. There is still a lot of material we want to add, we know that some material could be presented in a better way, and that there are loose ends that need to be tied up. There are also technical issues about the electronic medium we find unsatisfactory.

It is our modest hope, though, that we have included material that satisfies the majority of our readers, and that the electronic textbook is found useful. We encourage comments on content, shortcomings and errors to the address above.

Based on feedback from users and from our own experience in continued introduction in industrial contexts, we will incorporate new and better material in future versions of TIMe.

Future plans

Plans for the near future of the TIMe Electronic Textbook include:

- Tutorial on Z.109 SDL with UML
- Revise SDL Tutorial according to coming year 2000 version of Z.100
- Revise MSC Tutorial according to coming year 2000 version of Z.105

Revision table

Table 2-1: TIMe versions

Version	Date	What
4.0	July 1999	Second commercial version.
3.2	July 1998	Internal version for Ericsson.
3.1	December 1997	First commercial version from SINTEF.
3.0	November 1996	SISU L-2001-5 Final delivery from The SISU project (p.2-8), distributed to SISU Participants.
2.0	March 1996	Second version of Electronic textbook (distributed to SISU Participants).
1.0	September 1995	SISU L-2001-2 First version of electronic textbook. Demonstrated at SDL forum in Oslo.
Report	December 1994	SISU L-1112-6 Common Method Description, version 2.

Table 2-1: TIMe versions

Version	Date	What
Report	June 1994	SISU L-1112-5 Common Method Description, version 1.
Report	April 1994	SISU L-1112-4 First Draft of Integrated Methodology (in Norwegian).
Book	June 1993	ISBN 0-13-034448-6 Engineering Real Time Systems - An object-oriented methodology using SDL [24].

About SINTEF

SINTEF is an independent, not-for-profit research foundation based in Trondheim and Oslo, Norway. Our role is to encourage innovation and improve competitiveness in Norwegian industry and public administration. In doing so, we maintain close links with the technical Universities in Trondheim and Oslo, collaborating on projects, and sharing equipment and other resources.

SINTEF is not a publicly funded organisation. A very small part (less than 4%) of our income is from a public grant; most of our operating revenues arise from contract research and development work carried out for industry and the public sector in Norway and elsewhere.

With over 1800 employees and a turnover of NOK 1.4 billion, SINTEF is Scandinavia’s largest independent research organisation. It is organised into eight separate research institutes, covering all major scientific areas and industrial sectors.

SINTEF Telecom and Informatics is an institute in SINTEF. Our R&D work focuses on information and communications technologies - covering basic technologies as well as applications of these. We have departments specialising in computer science, systems engineering & telematics, distributed systems, radio & signal processing, and acoustics.

Refer to our website www.sintef.no for further information.

List of figures

TIMe activities, descriptions and languages 4
Methodology overview 5

List of definitions

Design oriented development 19
Language 19
Method 19
Methodology 19
Notation 20
Property oriented development 20

Design oriented development

An approach to system development where systems are understood and maintained mainly in terms of abstract design description in some notation or language.

Design oriented development is at a lower process maturity level than Property oriented development, but higher than implementation oriented development, where "the code documents the system".

Language

By a systems engineering language we mean a formal description technique (FDT). This means that not only the alphabet (notation) must be defined, but that both syntax (grammer) and semantics (meaning) of the language must be defined.

Examples of systems engineering languages are SDL, MSC, LOTOS, ESTELLE.

Contrast to Notation.

Method

A method is systematic way of producing some result.

In systems engineering a method provides guidelines for structuring and using descriptions in given notations.

Contrast to Methodology.

Methodology

A methodology is a collection of methods and guidelines for when and how to use them to produce a result.

In systems engineering most results take the form of descriptions expressed using some notation or language. A systems engineering methodology therefore prescribes a set of descriptions and associated methods.

A systems engineering methodology is used by an organisation in an attempt to achieve right quality, short lead times and low cost.

Notation

A systems engineering notation consists of symbols (an alphabet) that can be used to model or describe a concept or entity.

A notation is less formal than a Language, in that the syntax and/or the semantics are not formally defined.

Examples of notations are OMT, UML, ROOM, SA/SD, SADT.

Property oriented development

Property oriented development is characterized by an integration of:

- better product planning through focus on the early stages of system development, in particular domain analysis and requirements specification;
- emphasis on system families, evolution and reuse;
- formal expressions of required and provided properties;
- quality-by-construction through integration of methods for verification, validation and design synthesis.

Property oriented development is at a higher process maturity level than Design oriented development.