



LUND UNIVERSITY

The CACE project -- Steering Committee Meeting 1989-09-08

Mattsson, Sven Erik

1989

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Mattsson, S. E. (1989). *The CACE project -- Steering Committee Meeting 1989-09-08*. (Technical Reports TFRT-7437). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7437)/1-061/(1989)

The CACE Project -
Steering Committee Meeting, 1989-09-08

Sven Erik Mattsson

Department of Automatic Control
Lund Institute of Technology
November 1989

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		<i>Document name</i> Report	
		<i>Date of issue</i> November 1989	
		<i>Document Number</i> CODEN:LUTFD2/(TFRT-7437)/1-061/(1989)	
<i>Author(s)</i> Sven Erik Mattsson		<i>Supervisor</i>	
		<i>Sponsoring organisation</i> The Swedish Board of Technical Development	
<i>Title and subtitle</i> The CACE project - Steering committee meeting, 1989-09-08			
<i>Abstract</i> <p>This report contains documentation handed out to the participants of the steering committee meeting of the STU Computer Aided Control Engineering Programme (CACE) on September 8, 1989. The minutes of the meeting are also included.</p>			
<i>Key words</i> Computer Aided Control Engineering			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i>			<i>ISBN</i>
<i>Language</i> English and Swedish	<i>Number of pages</i> 61	<i>Recipient's notes</i>	
<i>Security classification</i>			

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

Preface

This report contains documentation handed out to the participants of the steering committee meeting of the STU Computer Aided Control Engineering Programme (CACE) on September 8, 1989. The minutes of the meeting are also included.

Table of Contents

Agenda	4
Overview of the Project	
<i>Status</i>	
Viewgraphs, Sven Erik Mattsson	5
<i>Published Papers, Conference Contributions and Reports</i>	8
Sven Erik Mattsson	
<i>Reports from some conferences</i>	15
Sven Erik Mattsson, Mats Andersson och Bernt Nilsson	
<i>Seminars and Visits</i>	16
<i>SEE and OMOLA</i>	
Viewgraphs and demonstration, Mats Andersson	27
Viewgraphs, Sven Erik Mattsson	28
<i>The application project</i>	
Viewgraphs, Bernt Nilsson	32
<i>Summary</i>	
Viewgraphs, Sven Erik Mattsson	35
A New Application Project	
Viewgraphs, Bernt Nilsson	39
Minutes of the Meeting, 1989-02-24	40
Minutes of the Meeting, 1989-04-06	47
Minutes of the Meeting, 1989-06-13	57
Minutes of the Meeting, 1989-09-08	61

CACE-projektet
Styrgruppssammanträde 1989-09-08

Dagordning

- 10.00 Formalia
- 10.15 Översikt av projektet
- 10.30 Rapporter från några konferenser
- 11.30 SEE och Omola
- 12.30 Lunch
- 14.00 Tillämpningsprojektet
- 14.20 Översiktlig sammanfattning av CACE-projektet med diskussion
- 15.20 Två nya tillämpningsprojekt
- 16.00 Avslutning

Översikt av projektet

1. Introduktion
2. Publicerade artiklar
3. Seminarier och besökare
4. Gästforskare

Publicerade artiklar 1989

1. Elmqvist, H. och S.E. Mattsson
"A Simulator for Dynamical Systems Using Graphics and Equations for Modelling"
Control Systems Magazine, Jan 1989
2. Mattsson, S.E.
"On Modelling and DAE Systems"
Simulation, No. 1, 1989
3. Nilsson, B., S.E. Mattsson, M. Andersson
"Tools for Model Developm. and Simul."
SAIS '89
Lund, May 7-9, 1989
4. Mattsson, S.E.
"Mod. of Interactions betw. Submodels"
ESM '89,
Rome, June 7-9, 1989
5. Andersson, M.
"An Object-Oriented Modeling Env."
ESM '89,
Rome, June 7-9, 1989

6. Mattsson, S.E.
"Concepts Supporting Reuse of Models"
Building Simulation '89
Vancouver, Canada, June 23-24, 1989
7. Nilsson, B.
"Structured Modelling of Chemical Processes with Control Systems"
Annual AIChE Meeting 1989
San Francisco, Nov 5-10, 1989
8. Brück, Dag M.
"Experiences of Object-Oriented Development in C++ and InterViews"
TOOLS'89
France, November 13-15, 1989
9. Andersson, M.
"An Object-Oriented Language for Model Representation"
IEEE CACSD'89
Tampa, Florida, December 16, 1989

Gästforskare

1. Dr. Wolfgang Kreutzer
Univ. of Canterbury, New Zealand
1985, Nov 18 - 1986, Jan 17
2. Prof. Mike Denham
Kingston Polytechnic
England
1986, Nov 6 - Dec 8
3. Prof. Dean Frederick
RPI, Troy, New York, USA
1987, May 17 - July 17
4. Dr. Wolfgang Kreutzer
Univ. of Canterbury, New Zealand
1987, Dec 1 - 1988, Jan 31
5. Prof. Doug Birdwell
Univ. of Tennessee
1988, May - June
6. Dr. Andre Tits
University of Maryland
1988, September

New forms of MMI

Expert system interfaces

Symbolic calculations

High level languages

Expert control

Repr. and visualization

Implementation lang.

Model development and simulation

IEEE CSS
CACSD '85 *

CPC III *

Appl. of AI
Southampton

* * * * *

ACC | SIAM
SERC/STU
UMIST

IEEE CSS
CACSD '86

*

SERC/STU
Cambridge

IFACAI
Swansea
|
ECAI-88
Munich
| * * * *
IFAC Beijing
|
USENIX C++
Denver
|
CADCS '88

ESM '89
Rome
| * * * *
SAIS '89
Lund
|
Building
Simulation '89
Vancouver

IEEE CSS
CACSD '89
Florida

**

AICHE '89
San Francisco

1985

1986

1987

1988

1989

WASHINGTON

DOD picks 22 technologies for conventional weapon R&D

The United States should focus its conventional (nonnuclear) defense research for fiscal 1989 on 22 technologies ranging from microelectronics to biotechnology, according to the first Critical Technologies Plan drawn up by the Department of Defense (DOD).

Every March from now on, as required by the Defense Authorization Act of Sept. 29, 1988, the DOD must present the Senate and House Committees on Armed Services with such a plan for developing a score or more process and product technologies.

The act, Public Law 100-456, further requires the undersecretary of defense for acquisition, in consultation with the assistant secretary of energy for defense programs, to select "the technologies most essential to develop in order to ensure the long-term qualitative superiority of United States weapon systems."

The selections and the background information included in the plan guide the DOD and Congress in apportioning defense spending among the budget categories of research, exploratory development, and advanced technology development.

One premise of the plan is that long-term defense superiority depends on a balanced national science and technology program with a range of mutually supportive technologies. In contrast, the DOD science and technology investment strategy aims at funding all relevant technologies.

The critical technologies chosen had to meet at least one of four performance and quality design criteria: the enhancement of conventional weapon system performance, the provision of new military capabilities, improved weapon system availability and dependability, and lower weapon system costs.

A select list

The 22 technologies picked and why they were judged critical are

listed below:

- Microelectronic circuits, whose increasing miniaturization will entail major weaponry modifications, whose inclusion of self-test circuitry could reduce maintenance, and whose methods of manufacture must turn out ICs that are highly reliable in combat.
- Gallium arsenide and other compound semiconductors, already important in microwave and millimeter-wave frequency devices and circuits, but with relatively undeveloped fabrication technologies, such as advanced epitaxy and thin-film deposition processes and equipment.
- Reliable and adaptable software, needed to reduce software development and maintenance costs (which far exceed hardware costs), plus software for large-scale distributed computation.
- Advances in parallel computer architectures, including the integration of processor elements for digital data, symbolic, and signal processing and the development of architectures and internal communication structures that will optimize performance.
- Simulation and modeling technology that can be applied to military systems to reduce design and production costs, improve performance and maintenance, train personnel, and enhance battlefield command and control.
- Integrated optics that will speed up computers and increase circuit reliability.
- Passive sensors to pick up electromagnetic energy in various ranges in order to enhance the detection, identification, and tracking of hostile weaponry.
- Phased arrays, used in radar systems that scan electronically by sequentially changing the phase of many small individual radiating elements, with near real-time complex signal-processing algorithms.
- Data fusion, or data processing, technology for countering adversarial stealth technology and assisting

target surveillance over wide areas of enemy territory.

- Signature control for altering radio or infrared radiation, acoustic waves, profiles, or other characteristics of weapon systems.
- Advances in air-breathing propulsion for aircraft, cruise missiles, land combat vehicles, ships, and future hypersonic systems, whether based on diesel, gas-turbine, or ramjet engines, to extend range and conserve fuel.
- High-power microwave technology methods for generating, maintaining, directing, and warding off intense microwaves. High-power beams may be able to defeat enemy weapon systems and interrupt enemy communications.
- Pulsed-power modulators, which by rapidly switching a beam on and off create a cannonade of short, intense bursts of directed energy, of use in high-energy lasers or high-power microwave weapons.
- Composite materials that withstand great heat and are strong and lightweight.
- Superconductivity technology of low-temperature metallic superconductors (for sensory and analog electronic devices) and high-temperature oxide superconductors, which are difficult to process and will require heavy R&D investment.
- Advances in automatic target recognition, biotechnology, computational fluid dynamics, fiber optics, hypervelocity projectiles, machine intelligence and robotics, and sensitive radars.

The unclassified version of the report is available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, Va. 22161; 703-487-4650. Defense contractors who are registered users may obtain copies of the report from the Defense Technical Information Center, Building 5, Cameron Station, Alexandria, Va. 22304; 202-274-7633.

—Katherine T. Chen

The CACE Project 1989-08-31
 Department of Automatic Control
 Lund Institute of Technology, Lund, Sweden

Published Papers, Conference Contributions and Reports

- ANDERSSON, M. (1989): "An Object-Oriented Modeling Environment," in G. Iazeolla, A. Lehman, H.J. van den Herik (Eds.): *Simulation Methodologies, languages and architectures and AI and graphics for simulation*, 1989 European Simulation Multiconference, Rome, June 7-9, 1989, The Society for Computer Simulation International, pp. 77-82.
- ANDERSSON, M. (1989): "Omola - An Object-Oriented Modelling Language," Report TFRT-7417, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ANDERSSON, M. (1989): "An Object-Oriented Language for Model Representation," IEEE CACSD'89, Tampa, Florida, December 16, 1989.
- ANON (1987): "Ramprogram Datorbaserade hjälpmedel för utveckling av styrsystem," *IT-aktuell*, 2:1987, 5-6.
- ÅRZÉN, K-E. (1986): "LISP—A One-Week Course," Report TFRT-7310, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ÅRZÉN, K-E. (1986): "Expert systems for process control," in Sriram, D. and R. Adey (Eds.): *Applications of Artificial Intelligence in Engineering Problems*, Proc. of 1st Int. Conf. on Applications of AI in Engineering Practice, Southampton University, U.K., April 1986, Springer-Verlag, pp. 1127-1138, Also available as report TFRT-7315, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ÅRZÉN, K-E. (1986): "Use of Expert Systems in Closed Loop Feedback Control," *Proc. ACC*, Seattle, USA, 1986, pp. 140-145, Also available as report TFRT-7320, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ÅRZÉN, K-E. (1986): "Kunskapsbaserade Regulatorer (Knowledge Based Controllers)," *SAIS '86*, The Swedish AI Society's Annual Workshop, Linköping, April 24-25, 1986.
- ÅRZÉN, K-E. (1986): "Reserapport—AAIEP 1986," Travel Report TFRT-8044, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ÅRZÉN, K-E. (1987): *Realization of Expert System Based Feedback Control*, Ph.D-thesis TFRT-1029, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ÅRZÉN, K-E. (1988): "An Architecture for expert system based feedback control," *Preprints of the IFAC Workshop Artificial Intelligence in Real-Time Control*, Swansea, September 20 - 23, 1988, pp. 15-20.
- ÅRZÉN, K-E. (1989): "An Architecture for Expert System Based Feedback Control," *Automatica*, Scheduled to appear in November 1989.

- ÅSTRÖM, K.J. (1985): "Computer Aided Tools for Control System Design—A perspective," in Jamshidi M. and C.J. Herget (Eds.): *Computer-Aided Control Systems Engineering*, North-Holland.
- ÅSTRÖM, K.J. (1986): "Auto-Tuning, Adaptation and Smart Control," in Morari and Mc Avoy (Eds.): *Proc Chemical Process Control—CPCIII*, CACHE, Elsevier, pp. 427–466.
- ÅSTRÖM, K.J., J.J. ANTON and K-E. ÅRZÉN (1986): "Expert Control," *Automatica*, **22**, No. 3, 277–286.
- ÅSTRÖM, K.J. and W. KREUTZER (1986): "System Representations," *Proceedings of the IEEE Control Systems Society Third Symposium on Computer-Aided Control Systems Design (CACSD)*, Arlington, Virginia, September 24–26, 1986, Also available as report TFRT-7330, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ÅSTRÖM, K.J., and S.E. MATTSSON (1987): "High-Level Problem Solving Languages for Computer Aided Control Engineering," Final Report 1987-03-31, STU project 85-4808, STU program: Computer Aided Control Engineering, CACE, Report TFRT-3187, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- BRÜCK, D.M. (1986): "Implementation of Graphics for Hibliz," Report TFRT-7328, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- BRÜCK, D.M. (1987): "Simplification of Expressions using Prolog," Report TFRT-7364, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- BRÜCK, D.M. (1987): "Design and Implementation of a Graphical Front-End," Report TFRT-7367, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- BRÜCK, D.M. (1987): "Implementation Languages for CACE Software," Final Report 1987-09-30, STU project 86-4047, STU program: Computer Aided Control Engineering, CACE, Report TFRT-3195, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- BRÜCK, D.M. (1988): "Modelling of Control Systems with C++ and PHIGS," *Proceedings of the USENIX C++ Technical Conference*, Denver, Colorado, October 17–20, 1988, pp. 183–192, Also available as report TFRT-7400, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- BRÜCK, D.M. (1989): "Experiences of Object-Oriented Development in C++ and InterViews," Report TFRT-7418, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- BRÜCK, D.M. (1989): "Scones — An Interactive Block Diagram Editor for Simnon," Report TFRT-7423, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- DENHAM, M.J. (1987): "Knowledge Representation in Systems Modelling," Report TFRT-7365, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

- ELMQVIST, H. and S.E. MATTSSON (1986): "A Simulator for Dynamical Systems Using Graphics and Equations for Modelling," *Proceedings of the IEEE Control Systems Society Third Symposium on Computer-Aided Control Systems Design (CACSD)*, Arlington, Virginia, September 24-26, 1986, pp. 134-139.
- ELMQVIST, H. and S.E. MATTSSON (1989): "Simulator for Dynamical Systems Using Graphics and Equations for Modeling," *IEEE Control Systems Magazine*, 9, 1, 53-58.
- FREDERICK, D.K. (1987): "An Introductory Study of a Window-Based Environment for Simnon on the SUN Workstation," Report TFRT-7366, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- GRANBOM, E. and T. OLSSON (1987): "VISIDYN - Ett program för interaktiv analys av reglersystem," MSc Thesis TFRT-5375, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- HOLMBERG, U. (1986): "Some MACSYMA Functions for Analysis of Multivariable Linear Systems," Report TFRT-7333, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- HOLMBERG, U., M. LILJA and B. MARTENSSON (1986): "Integrating Different Symbolic and Numeric Tools for Linear Algebra and Linear Systems Analysis," Report TFRT-7338, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, Presented at SIAM Conference on Linear Algebra in Signals, Systems and Control, August 12-14, 1986, Boston, Massachusetts, USA.
- HOLMBERG, U., M. LILJA and S.E. MATTSSON (1987): "Combination of Symbolic Manipulation and Numerics," Final Report 1987-03-31, STU project 85-4809, STU program: Computer Aided Control Engineering, CACE, Report TFRT-3186, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- HOLMBERG, U., M. LILJA and B. MARTENSSON (1988): "Integrating Different Symbolic and Numeric Tools for Linear Algebra and Linear Systems Analysis," in B.N. Datta, C.R. Johnson, M.A. Kaashoek, R.J. Plemmons and E.D. Sontag (Eds.): *Linear Algebra in Signals, Systems, and Control*, Proceedings of the Conference on Linear Algebra in Signals, Systems and Control, Boston, Massachusetts, USA, August 12-14, 1986, SIAM, Philadelphia, pp. 384-400.
- JOHANSSON, M. (1988): "Interaktiv plottning av mätdata i flera dimensioner," (Interactive plotting of measurement data in several dimensions), MSc thesis TFRT-5390, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- JEPPSSON, U. (1988): "An Evaluation of a PHIGS Implementation for Full Graphics Control Systems," MSc thesis TFRT-5389, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- KREUTZER, W. and K.J. ÅSTRÖM (1987): "An Exercise in System Representations," Report TFRT-7369, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- LARSSON, J.E. (1984): "An Expert System Interface for Idpac," MSc

- thesis TFRT-5310, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- LARSSON, J.E. (1989): "Knowledge-Based Frequency Response Analysis," in Larsson, J.E. (Ed.): *Proceedings of the SAIS '89 Workshop*, AILU—The AI Group in Lund, Lund University and Lund Institute of Technology.
- LARSSON, J.E. and K.J. ÅSTRÖM (1985): "An Expert System Interface for Idpac," Report TFRT-7308, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, Presented at CACSD '85, The 2nd IEEE Control Systems Society Symposium on Computer-Aided Control System Design (CACSD), Santa Barbara, California, March 13–15, 1985.
- LARSSON, J.E. and P. PERSSON (1986): "Ett expertsystemsnitt för Idpac (An Expert System Interface for Idpac)," *SAIS '86*, The Swedish AI Society's Annual Workshop, Linköping, April 24–25, 1986.
- LARSSON, J.E. and P. PERSSON (1986): "Knowledge Representation by Scripts in an Expert Interface," *Proceedings of the 1986 American Control Conference*, Seattle, June 1986, Also available as report TFRT-7332, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- LARSSON, J.E. and P. PERSSON (1987): "An Expert System Interface for Idpac," Licentiate Thesis TFRT-3184, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- LARSSON, J.E. and P. PERSSON (1987): "Ett intelligent gränssnitt för systemidentifiering (An Intelligent Interface for System Identification)," *Proceedings of the SAIS'87 Workshop*, The Swedish AI Society's Annual Workshop, Uppsala, May 18–19, 1987.
- LARSSON, J.E. and P. PERSSON (1987): "Experiments with an Expert System Interface," Final Report 1987-09-30, STU project 85-3042, STU program: Computer Aided Control Engineering, CACE, Report TFRT-3196, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- LARSSON, J.E. and P. PERSSON (1987): "The (ihs) Reference Manual," Report TFRT-7341, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- LARSSON, J.E. and P. PERSSON (1987): "A Knowledge Database for System Identification," Report TFRT-7342, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- LARSSON, J.E. and P. PERSSON (1988): "An Intelligent Help System for Idpac," *Proceedings of ECAI-88*, European Conference on Artificial Intelligence, Munich, August 1 – 5, 1988, pp. 119–123.
- LARSSON, J.E. and P. PERSSON (1988): "The knowledge database used in an expert system interface for Idpac," *Preprints of the IFAC Workshop Artificial Intelligence in Real-Time Control*, Swansea, September 20 – 23, 1988, pp. 107–112.
- LILJA M. (1986): "Some SISO Transfer Function Facilities in CTRL-C," Report TFRT-7325, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

- MATTSSON, S.E. (1985): "New Forms of Man-Machine Interaction," Status Report 1985-09-30, STU project 84-5069, STU program: Computer Aided Control Engineering, CACE, Report TFRT-7293, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. (1986): "On Differential/Algebraic Systems," Report TFRT-7327, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. (Ed.) (1987): "Computer Aided Control Engineering, CACE," Proceedings of a full-day seminar, Stockholm, March 25, 1987 Report TFRT-7359, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. (Ed.) (1987): "Programplan för ramprogrammet Datorbaserade hjälpmedel för utveckling av styrsystem (Computer Aided Control Engineering, CACE)," Final Report TFRT-3193, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. (1987): "Hibliz ger enkel och klar modellbeskrivning på varje nivå," *IT-aktuellt*, 2:1987, p. 11.
- MATTSSON, S.E. (1987): "Representation and Visualization of Systems and Their Behaviour," Final Report 1987-09-30, STU project 86-4049, STU program: Computer Aided Control Engineering, CACE, Report TFRT-3194, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. (1988): "On Model Structuring Concepts," *Preprints of the 4th IFAC Symposium on Computer-Aided Design in Control Systems (CADCS)*, August 23-25 1988, P.R. China, pp. 269-274.
- MATTSSON, S.E. (1989): "On Modelling and Differential/Algebraic Systems," *Simulation*, 52, No. 1, 24-32.
- MATTSSON, S.E. (1989): "Modeling of Interactions between Submodels," in G. Iazeolla, A. Lehman, H.J. van den Herik (Eds.): *Simulation Methodologies, languages and architectures and AI and graphics for simulation*, 1989 European Simulation Multiconference, Rome, June 7-9, 1989, The Society for Computer Simulation International, pp. 63-68.
- MATTSSON, S.E. (1989): "Concepts Supporting Reuse of Models," *Proceedings of Building Simulation '89*, Vancouver, June 23-24, 1989, The International Building Performance Simulation Association, pp. 175-180.
- MATTSSON, S.E. and M. ANDERSSON (1989): "A kernel for System Representation," Report TFRT-7429, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. and K.J. ÅSTRÖM (1986): "The CACE Project— Steering Committee Meeting 2," Report TFRT-7321, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. and K.J. ÅSTRÖM (1986): "The CACE Project— Steering Committee Meeting 3," Report TFRT-7322, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. and K.J. ÅSTRÖM (1987): "The CACE Project— Steering Committee Meeting 4," Report TFRT-7343, Department of Automatic

Control, Lund Institute of Technology, Lund, Sweden.

- MATTSSON, S.E. and K.J. ÅSTRÖM (1988): "The CACE Project— Steering Committee Meeting, 1987-11-25," Report TFRT-7375, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. (1988): "The CACE Project— Steering Committee Meeting, 1988-06-01," Report TFRT-7395, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E. (1989): "The CACE Project— Steering Committee Meeting, 1988-11-23," Report TFRT-7412, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E., K.J. ÅSTRÖM, D.M. BRÜCK and T. SCHÖNTHAL (1987): "A Trip to the University College Swansea and the Central Electric Generating Board, Gloucester," Travel Report TFRT-8045, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MATTSSON, S.E., H. ELMQVIST and D.M. BRÜCK (1986): "New Forms of Man-Machine Interaction," Final Report 1986-09-30, STU project 84-5069, STU program: Computer Aided Control Engineering, CACE, Report TFRT-3181, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MARTENSSON, B. (1986): "CODEGEN—Automatic Simmon Code Generator for Multivariable Linear Systems," Report TFRT-7323, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MARTENSSON, B. (1986): "Automatic T_EX Code Generation from Macsyma and CTRL-C," Report TFRT-7334, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MARTENSSON, B. (1986): "Hcopy2PS-Hcopy to PostScript filter," Report TFRT-7335, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MARTENSSON, B. (1986): "Experiences With Computerized Document Preparation Tools," Report TFRT-7336, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MARTENSSON, B. (1987): "Pascal and Fortran Systems in VAX/VMS Simmon – A Cookbook," Report TFRT-7351, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- MARTENSSON, B. (1987): "Documentation of MacEQ2T_EX, DVILW, and Hcopy2PS," Report TFRT-7352, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- NILSSON, B. (1986): "LQG-I/O—A CTRL-C library for LQG design," Report TFRT-7329, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- NILSSON, A. (1988): "Object-oriented Graphics for the Future Instrument Panel," MSc thesis TFRT-5382, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- NILSSON, B. (1987): "Experiences of Describing a Distillation Column in Some Modelling Languages," Report TFRT-7362, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

- NILSSON, B. (1989): "Structured Modelling of Chemical Processes with Control Systems," Annual AIChE Meeting 1989, San Francisco, Nov 5-10, 1989.
- NILSSON, B., S.E. MATTSSON and M. ANDERSSON (1989): "Tools for Model Development and Simulation," in Larsson, J.E (Ed.): *Proceedings of the SAIS '89 Workshop*, AILU—The AI Group in Lund, Lund University and Lund Institute of Technology.
- VALLINDER, P.A. (1988): "Some Methods for Tearing of Differential/Algebraic Systems," MSc thesis TFRT-5384, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Rapporter från några konferenser

1. SAIS'89, Lund
2. International Conference on System Developments & Factories
3. ESM'89, Eur. Simulation Multiconference
4. Building Simulation '89
5. Summer School on User Interfaces, Tampere

Seminars and Visits

November 1988 – August 1989

Sven Erik Mattsson

Department of Automatic Control
Lund Institute of Technology
Lund, Sweden

This is a list of seminars and external contacts the Department of Automatic Control, Lund Institute of Technology has had during the period November 1988 – August 1989 which are of interest for the CACE project. The list includes visits to the department and visits of the staff to companies and other universities, as well as participation in conferences, symposia, workshops, courses etc.

Our visitors are normally given a presentation of our department and our research, as well as live demonstrations of our packages for CACE, so this is not explicitly mentioned in the list below.

1988

- Nov 24 Professor Ed Sowell, California State University, Fullerton and Per Sahlin, ITM, Stockholm visited the department. Their main interest are building simulation. Professor Sowell gave a seminar titled "Simulation Analysis Kernel (SPANK) – and object oriented environment" and Per Sahlin gave a seminar titled "Desired features in model and simulation environments for the building industry".
- Nov 25 Ola Dahl presented his system for implementing controllers. It converts a controller given in Simmon to a Modula-2 program.
- Dec 2 Nevile Rees from Sidney, Australia gave a seminar titled "On the Mechanization of Thought Processes".
- Dec 14–15 Sven Erik Mattsson participated in the workshop on "Future Research Needs in CACSD", Cambridge 14 & 15 December 1988. It was arranged by the Science and Engineering Research Council, SERC in UK. People from industry and universities in UK participated. Sven Erik Mattsson was invited to give a presentation "Future Modelling and Simulation Environment".

1989

- Jan 13 Karl Johan Åström gave a seminar titled "Neural networks".
- Feb 3 Dag Brück gave a seminar titled "Scones - an interactive tool for creating Simmon's connecting systems". Scones allows you to define Simmon's connecting systems as block diagrams. Scones is implemented in C++ and InterViews. Scones was developed in the CACE project to get experiences of using C++ and InterViews.
- March 2 Torbjörn Olsson and Klas Lysen, Kamyr AB, Karlstad visited the department. They are working with process control of continuous digesters for dissolving pulp and paper pulp. Torbjörn Olsson has been the president of MODO Chemetics. They gave a seminar titled "Control of a continuous digester".
- March 7 About 10 persons from TeleLOGIC, Malmö visited the department. Half of the group are members of TeleLOGIC's KBS group. Mats Petterson gave a seminar and presented TeleLOGIC in general and their KBS activity.
- March 22 Dr Nigel Baker, Applied Dynamics International, the English office visited the CACE group to learn about the CACE project. He had read Sven Erik Mattsson's paper "On modelling and differential/algebraic systems" in Simulation.
- Dr Baker has a solid background in simulation. He has participated in many English defense projects on modelling and simulation of missiles. Inspired by Francois Cellier's COSY he developed the simulation language SYSMOD for combined continuous time and discrete event simulation. He wanted to support differential/algebraic equations in SYSMOD, but the English Department of Defense did not allow him to do that. They supported at this time also at the development of the simulation language TSIM, which is a CSSL derivative. TSIM is rather commonly used in the UK today.

Applied Dynamics, International has its main office in Ann Arbor, Michigan, USA. They have manufactured analog computers. Today their main product SYSTEM 100 for in the loop, real-time simulations. It runs from five to fifty times faster than general-purpose computers. The modelling language is called ADSIM and is a CSSL derivative.

Dr Baker liked our ideas and meant that we are on the right way. He did not know about something similar, but thought that we are in the front line.

Tomas Schönthal demonstrated Simmon. Dr Baker liked it and was impressed that over 500 licenses had been sold.

- March 31 Tomas Schönthal gave an overview of some German CACE software for PCs. Tomas participated in the workshop "Regelungstechnische Programmpakete für den IBM-PC und kompatible Personalcomputer unter MS-DOS" March 2-3, Düsseldorf. It was arranged by Verein Deutscher Ingenieure, VDI, Bildungswerk. Tomas presented and demonstrated Simmon.
- Apr 7 Sten Bergman, ABB, Västerås gave a seminar titled "VLSI system construction".
- Apr 14 Professor Torsten Bohlin, KTH, gave a seminar titled "On the foundations of system identification".
- May 7-9 The Swedish Artificial Intelligence Society, SAIS arranges an annual workshop. This year the workshop SAIS '89 was arranged in Lund. Jan Eric Larsson was the local organizer of the workshop. About 60 persons attended the workshop. Bernt Nilsson made a presentation of the CACE project.
- May 9-11 Mats Andersson attended the "International Conference on System Development Environments & Factories" in Berlin. The topic was integrated environments and tools supporting software development. The conference was arranged by a number of large project groups, which presented their projects and results. Most of the projects presented belongs to the EUREKA or ESPRIT program. See Appendix A.
- May 26 Professor Torkel Glad, LiTH gave a seminar titled "Differential algebra and modelling". This area is of interest for the CACE project since there are interesting ideas for symbolic manipulation and analysis of differential algebraic systems.
- June 7-9 Sven Erik Mattsson and Mats Andersson participated in ESM'89, the 1989 European Simulation Multiconference, June 7-9, 1989, Rome, Italy and presented two papers: "An object-oriented modelling environment" (Andersson) and "Modelling of interaction between submodels" (Mattsson). See Appendix B.
- June 14 Visit by J. Grievink and H. Tulleken, Process Control and Optimization, Department of Mathematics and Systems Engineering, Koninklijke/Shell-Laboratorium (K.S.L.A.), Amsterdam. J. Grievink gave a seminar and presented the process control research at K.S.L.A.
- June 16 Claes Ryttoft and Henrik Pålson, ABB, Ideon presented their project "Future man-machine interaction".

- June 20 Ralph Myrnäs presented his Master thesis project "A modification of the Barsebäck simulator". The simulator is a compact simulator of the nuclear power plant at Barsebäck for operator training. The simulator is implemented using Simnon.
- June 21 Visit by Professor John Baras and Andre Tits, Systems Research Center at the University of Maryland. We gave a brief presentation of the CACE project and they presented their optimization tool CONSOLE. Then we discussed the possibility of integrating our two systems.
- June 23-24 Sven Erik Mattson participated in Building Simulation '89, Vancouver, Canada. The conference was arranged by The International Building Performance Simulation Association, IBPSA. IBPSA promotes the science of building performance simulation in order to improve the design, construction, operation, and maintenance of all types of buildings. IBPSA's international membership includes architects, engineers, building managers, academics, software developers, and government representatives concerned with building performance. Sven Erik Mattsson presented his paper "Concepts Supporting Reuse of Models". He learned about IBPSA and Building Simulation '89 in November 1988 when Professor Sowell visited the department. He found our work interesting and encouraged us to submit a paper. Professor Sowell presented a proposal by him and Per Sahlin, ITM for a neutral format for building simulation models to allow users to share models. This proposal is inspired and influenced by the results of the CACE project.
- June 26 - July 1 Mats Andersson attended the "Summer School on User Interfaces" in Tampere, Finland. The course was organized by the Finnish Society for Computer Science and the University of Tampere. It was given in six days of seminars by five lecturers: Richard A. Bolt from MIT, Cambridge, Massachusetts; Marc H. Brown, DEC's System Research Center in Palo Alto, California; James D. Foley from the George Washington University, Washington D.C.; Brian W. Kernighan, AT&T Bell Laboratories, Murray Hill, New Jersey; and Ben Shneiderman from the University of Maryland. The course covered broad aspects of human - computer interaction. Among the topics were: Methodology and guidelines for designing user interfaces, interaction styles, algorithm animation, hyper text, and UIMS (User Interface Management Systems).

Appendix A — En reseberättelse från "International Conference on System Development Environments & Factories" 9–11 Maj 1989, i Berlin.

Mats Andersson
 Department of Automatic Control
 Lund Institute of Technology, Lund, Sweden

1. Inledning

Inom CACE-projektet besökte jag förra veckan en konferens i SD E&F i Berlin. Konferensen handlade om integrerade miljöer med verktyg och stöd för utveckling av mjukvara, och arrangerades av ett antal stora projektgrupper och konsortier, främst inom EUREKA- och ESPRI-programmen.

Främst handlade konferensen om stöd för att driva stora mjukvaruprojekt. Det finns ett stort antal olika verktyg för programvaruutveckling som t.ex. texteditorer, syntaxbaserade editorer, kompilatorer, avlusare och databaser för att hålla reda på versioner och dokument. En del verktyg är språkberoende medan andra är generella. Problemet är att man vill integrera dessa verktyg i en gemensam enhetlig miljö. Inom området satsas det stort, speciellt här i Europa, i form av flera stora sameuropeiska projekt inom EUREKA och ESPRI och man försöker enas om vissa standards.

Programutveckling har tidigare betraktats enligt den s.k. vattenfallsmodellen. Ett projekt går igenom faserna problem definition, specifikation, implementering och underhåll i en rak följd och inga återhopp sker. Vill man utveckla verktyg som stöder samtliga faser i programutvecklingen behöver man en bättre modell av processen som i praktiken är mycket mer komplicerad än vattenfallsmodellen. Man sysslar alltså ganska mycket med att försöka modellera och formalisera programutveckling betraktat som en dynamisk process. Detta är mycket svårt!

En CASE-miljö (CASE = Computer Aided Software Engineering) är baserad på någon form av modell av utvecklingsprocessen som kan vara mer eller mindre formaliserad. Dessutom karakteriseras den av:

- En metodologi
- Ett (eller flera) programmeringsspråk
- Verktyg (som editorer, kompilatorer, avlusare, m.m.)
- Stöd för projektstyrning (Management support).

För programutvecklingsmiljöer kan man urskilja tre generationer:

1. Stöd för programmering i det lilla formatet såsom syntaxeditorer och liknande. (En bra översikt över denna generation programmeringsstöd finns i Barstow, Shrobe, Sandewall "Interactive Programming Environments").
2. Stöd också för andra faser i programutvecklingen såsom specifikation, dokumentation och underhåll.
3. Man betraktar programutveckling som en process och ger stöd för styrning av denna process. Metodologi och organisation av resurser innefattas av processmodellen. Begreppet "Software factory" införs. Det var mest denna senaste generation som diskuterades vid konferensen.

2. Några intressanta projekt och presentationer

2.1 ESF

ESF (Eureka Software Factory) är ett konsortium av ett antal Europeiska företag (ca 10 tror jag), där bl.a TeleLogic ingår, som utvecklar och marknadsför CASE programvara. Deras strategi presenteras i "The ESF Vision of a Software Factory" av Christer Fernström (Cap Sesa Innovation) och Lennart Ohlsson (TeleLogic) (båda har tidigare varit på Digital- och DatorTeknik, LTH).

Här driver man begreppet "Software Factory" till sin spets. Programutveckling betraktas som en process som ska styras och denna styrning kan delvis automatiseras. Den enskilde programmeraren tillhör processen och blir delvis underkastad automatisk styrning (Scener ur Chaplins "Moderna Tider" känns aktuella än någonsin.)

2.2 PCTE

PCTE (Portable Common Tool Environment) startade som ett ESPRI-projekt 1983 och har sedan dess mognat en hel del. PCTE är ett försök att skapa en standardiserad dator och operativsystem-oberoende miljö i vilken CASE-verktyg kan integreras. PCTE kan ses som ett skal utanpå operativsystemet och är främst en angelägenhet för dem som utvecklar nya CASE-verktyg.

Ett antal kommersiella implementeringar av PCTE för UNIX och VMS finns eller väntas inom kort och ett antal verktyg baserat på PCTE håller på att utvecklas. Med PCTE följer i allmänhet också bibliotek med kod som behövs i de flesta CASE-verktyg för t.ex användarinteraktion och liknande.

Om PCTE kan man läsa i "The International PCTE Initiative" av Ian Campbell och i diverse broschyrer.

2.3 ESSI

ESSI (The European System & Software Initiative) är ett projekt som just startats inom ESPRI II. Projektet är på ca 4000 manår och ska vara i 4 år. Syftet är att öka produktiviteten vid mjukvaruproduktion genom att framförallt föra ut ny teknologi från ett antal ESPRI I-projekt till IT-industrin.

Jag har ingen skriven information om projektet utan nämmer det här som ett exempel på vilka stora projekt man hanterar inom ESPRI-programmet.

2.4 Leonardo

Leonardo är ett amerikanskt projekt som drivs av MCC i Austin, Texas, ett konsortium av ca 20 hård- och mjuk-varuföretag. Leonardo är en CASE-miljö som integrerar olika verktyg och som betonar betydelsen av återanvändning av kod. Systemet bygger objektorientering och på ett s.k. "Meta Level Integration Language". Man kan läsa "A Vision of Leonardo" av Ted J. Biggerstaff!!

Ett citat: " ... the main influences of Leonardo will be ... an evolution toward a computer-based culture that captures corporate wisdom, ... " – Du sköna nya värld!

2.5 EPOS

EPOS (Expert System for Program "Og" System development) är ett mindre projekt vid Norges tekniska högskola i Trondheim. EPOS innehåller en kärna, verktyg och expertsystem för processadministration och konfigureringsystem. (Konfigureringsystem innebär i detta sammanhang att plocka ihop rätt moduler och länka dessa för kundspecifika leveranser av mjukvara).

Artikeln "Design of the kernel EPOS Software Engineering Environment" av Reidar Conradi m.fl. är läsbar och förhållandevis konkret.

2.6 ECMA – TC33 User Interfaces

ECMA (European Computer Manufacturers Association) är ett konsortium av europeiska datortillverkare som gör viss grundläggande forskning och diskuterar standards. "ECMA TC 33 User Interfaces Technical Assessment Report" diskuterar grafik och fönstersystem, främst X-windows och interaktionspaket ovanpå X. Man presenterar en referensmodell för användargränssnitt i sju nivåer. Principer för interaktion och portabilitet diskuteras.

2.7 AIMS

AIMS (Aerospace Intelligent Management and Development Tool for Embedded Systems) är ett projekt på Messerschmitt Bölkow Blohm GmbH i Västtyskland. Detta är det enda projekt som speciellt diskuterar problemen med "embedded systems" med realtidskrav och extremt höga tillförlitlighetskrav.

Här krävs datorstöd även för att administrera testkörningar och simuleringar av systemen. Projektet presenteras väldigt allmänt men artikeln innehåller en del intressant uppgifter om produktivitet vid konstruktion av denna typ av mjukvara (1 rad källkod per person och dag) och hur mycket man förväntas höja denna produktivitet med hjälp av bättre verktyg.

3. Allmänna intryck

Om man bortser från ren projektadministration så har de projekt, problem och lösningar som diskuterats vid konferensen stora likheter med vårt CACE-projekt. Betoningen ligger alltså på integrering av olika verktyg och återanvändning av kod/modeller. Man betonar också att det är viktigt med verktygslådor/miljöer som stöder hela utvecklingsprocessen från specifikation via implementering till utprovning och underhåll.

Objektorienterat tänkande kommer mer och mer, speciellt vill man använda objektorienterade databaser för att representera modeller och objekt med komplicerade relationer.

Implementeringsspråk har en underordnad betydelse men generellt tycks gälla: Ada för militärer, Modula-2 för universitet och C för industrin.

Ett initiativ som PCTE (se ovan) som håller på att bli standard vore trevligt för CACE.

Appendix B — Travel Report from ESM'89

Sven Erik Mattsson and Mats Andersson
 Department of Automatic Control
 Lund Institute of Technology, Lund, Sweden

Introduction

This report is a travel report from ESM'89, the 1989 European Simulation Multiconference, June 7–9, 1989, Rome, Italy. First a short introduction of the Society for Computer Simulation (SCS) International, which organized the conference, is given.

SCS International

The Society for Computer Simulation (SCS) International is a non-profit organization which aim is to provide means for exchanging information about simulation methods (research, hardware and software) and use of simulation in various disciplines. The history of SCS starts in 1952 when users of simulation in California organized a professional society called Simulation Councils. The SCS International is now a worldwide organization with many affiliations. In Scandinavia there is the Scandinavian Simulation Society, SIMS.

SCS International publishes books and the two journals *Simulation* and *Transactions of The Society for Computer Simulation*.

SCS International organizes each year a number of conferences on simulation methodologies and applications. The largest one is the Summer Computer Simulation Conference. It is held in the U.S. or Canada. The 1989 Summer Computer Simulation Conference will take place in Austin, Texas, July 24–27. Another conference is the SCS Multiconference, which is held in San Diego each winter. A third, rather new, major conference is the European Simulation Multiconference (ESM). ESM'89 took place in Rome, Italy on June 7–9. ESM'90 will take place in Erlangen (Nurenburg), West Germany on June 11–13.

ESM'89

The ESM'89 was a three day conference with four parallel sessions. The topics were

1. Simulation methodologies, languages and architectures
2. AI & graphics for simulation
3. Simulation applied to
 - a. manufacturing
 - b. energy and environmental studies
 - c. electronics and computer engineering

The conference proceedings are published in two volumes, where one volume covers the topics 1 and 2 and the other covers the applications. The number of participants was in the range 100-150.

Below we will mention some issues that we found particularly interesting.

Number crunching

Professor Guiseppa lazeolla, Rome gave the keynote speech "A few remarks on cost/performance relationships in parallel computer architectures".

Hardware and distributed parallel simulation was a major topic at the conference. The Applied Dynamics International gave a couple of presentations and live demonstrations of their SYSTEM 100 for in the loop, real-time simulations. The SYSTEM 100 runs from five to fifty times faster than general-purpose computers. The modelling language is called ADSIM and is a CSSL derivative.

The use of AI in Simulation

Dr Marc Fox, Intelligent Systems Laboratory Robotics, Robotics Institute, Carnegie Mellon University, Pittsburgh, USA gave a state-of-the-art presentation. Dr Fox is known as a pioneer in the field of AI and in the development of knowledge based simulation environments. His presentation was titled "Role of AI in Simulation: Present and Future Directions" but restated as "What has AI contributed to simulation and what will it contribute". He first described Simulation Craft which is a environment for knowledge based simulation from the Carnegie group. It contains a model of the simulation activity. The user defines the goal of his task (by selecting from menus). The system then guides and supervises the user.

Dr Fox pointed that the contributions from AI to simulation are in many cases indirect; methods and hardware that were used early in AI because of its complexity are now used in simulation: workstations, high interactivity, computer graphics, animation, object orientation, direct manipulation etc. A point of special interest for us is that he mentioned definition of event behaviour using rules. It is easier than coding in Fortran. It allows partial models of system behaviour. The user is not forced to specify the complete behaviour, but he may only specify the situations when he has knowledge. Dr Fox did not know the impact of qualitative simulation.

He mentioned two future directions. First, much can be done to improve the user interface. He said that natural language remains a promising technology. Smart graphics can be used to compare and display the differences of multiple simulations. As the second area of great interest he mentioned our favourite: ontologies or identification of entities, their properties and the relationships among them. He pointed out that a standard of representing knowledge is needed and outlined a number of concepts, which agree well with our views and our research.

Continuous simulation

Dr. Ralph Huntsinger (former president of SCS), California State University, Chico gave a tutorial seminar "A survey of modern continuous systems simulations languages". He pointed out that the most used languages like CSSL IV, CSMP, DSL/V, ACSL, EASY5 are very similar. The price and hardware requirements may vary. He mentioned that ESL (European Simulation language) and COSMOS are advanced versions of CSSL IV, since they have discontinuity and discrete event facilities. DESIRE by Korn has an interesting tradeoff between interactivity and efficiency. The dynamic parts of the model are compiled while the rest is interpreted.

The examples which he used to illustrate the languages were very small and simple. He did not discuss the simulation of large systems and model structuring.

Action: Send documentation about Simnon as well as a demo floppy to Dr. Ralph C. Huntsinger, Department of Computer Engineering, California State University, Chico, California 95929-0410.

Object orientation

There was a special invited session on the topic of object-oriented modelling. Of special interest for us was the paper "Conceptual tools for modelling complex objects" by Braspening et al, the University of Limburg, The Netherlands. They thought that the traditional object-oriented concepts are too restrictive and proposed a new set of concepts. Their application is CASE (Computer Aided Software Engineering). However, the paper is rather abstract and examples are needed to better understand their proposal. Unfortunately, their internal reports are in Dutch, but they have planned to write more papers in English. We think that it is of mutual interest to keep in touch with them.

Another interesting presentation was "Developments in an object oriented environment for digital signal applications" by Zobel and Cummings, the University of Manchester, UK. They had found that it is in many cases not obvious whether operations should be implemented as methods or as special objects. For example, should FFT be a method of signals or a special object (machine)? They were going to pursue both approaches to get a deeper insight and experiences.

Environments and graphics

Some papers focused on environments of integrated tools supporting simulation, analysis and design. This is related to our work. T. Ketonen from Nokia Research Center, Helsinki, presented the paper "An Intelligent Design and Simulation Environment" which reports on an implementation of an environment supporting design of high frequency electronic circuits. This environment, called RFT, combine tools for numeric and symbolic analysis, simulation and a graphical interface. The environment also contains knowledge of how to use the different tools in a design process (IHS).

R. Ruzicka (Technical University of Vienna, Austria) presented an environment for SIMUL-R – a simulation system for continuous and discrete models. The environment has a graphic interface with information zooming on models. It is also possible to create real time graphic animations of the simulation and to scan photographic pictures and use them in the animation – a feature that is nice for demonstrations. The system is a commercial product.

Our contributions

We had two papers: "An object-oriented modelling environment" (Mats Andersson) and "Modelling of interaction between submodels" (Sven Erik Mattsson). They were well received and lead to new acquaintances which can be of mutual usefulness.

Dr Bernd Schmidt, University of Erlangen, West-Germany found our ideas very interesting ("one of the best contributions at the conference"). His speciality is discrete-event simulation and he has developed a description language SIMPLEX-MDL. It is object-oriented, hierarchical and incorporates concepts from logical programming. Dr Schmidt is a member of the organization committee of ESM'90 and he asked us to submit contributions.

Dr Wayne Ingalls (IJK Inc., Maple Valley, Washington) found our ideas interesting, but pointed out that it is not easy to check that models from a model library is used correctly. A basic difficulty is how to check that the user does not put together models so that a state is represented twice in the model. If such two states are initialized consistently, then everything is fine in the beginning of a simulation, but after some time the model behaves strangely because of drift due to the limited numerical accuracy. You may note that a

component based decomposition approach ought to be less exposed to errors in this respect than a functional decomposition. The problem is closely related to the topic of minimal realizations and controllability, since it is reasonable to allow a user to inspect all variables. The theory of differential algebra and symbolic manipulation could be useful tools.

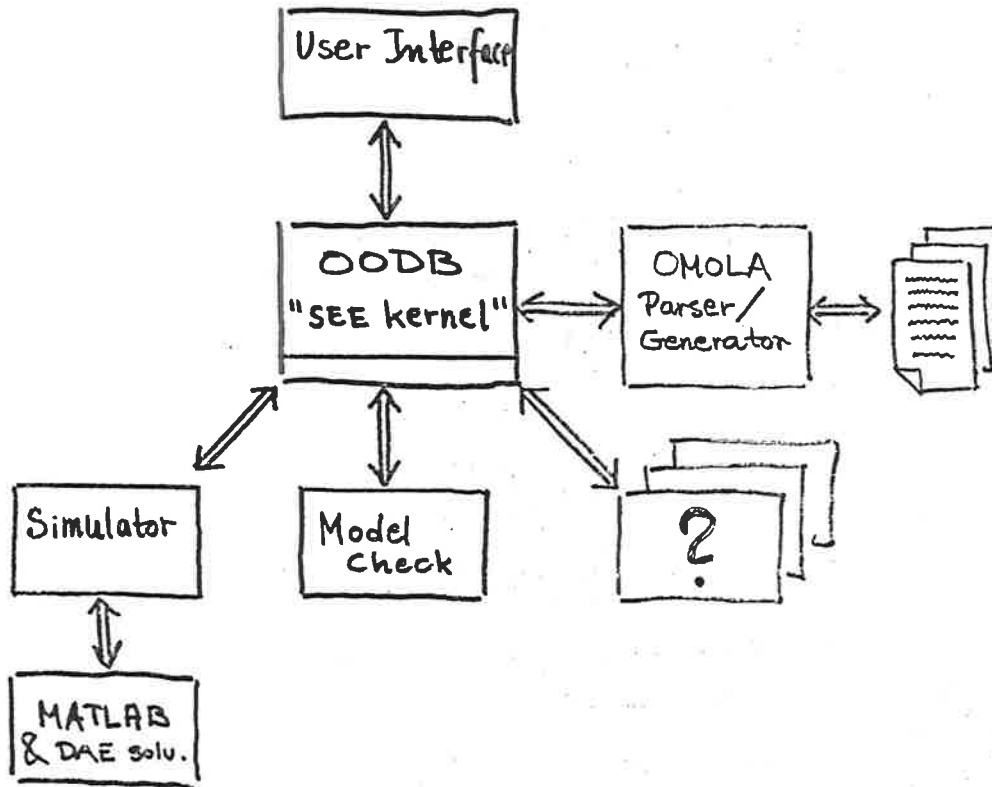
Action: Consider if we should start research to develop methods to detect and eliminate “redundant states”.

Conclusion

For us ESM'89 was a successful conference:

1. We met a different community of simulation users.
2. We became acquainted with several persons with whom we can exchange ideas and experiences.
3. Our presentations were well received and the organizers of ESM'90 asked us to submit contributions.

SEE Architecture



SEE User Interface

* Library browser

* Model Display

* Editors

- Text editor for primitive realizations
- Graphic editor for struct. models
- Terminal attribute form

Kompilering för simulering

1. Generera och samla ihop ekvationerna
2. Analysera och manipulera
3. Generera kod för simulering

DASSL

DASSL löser problem av typen

$$g(t, \dot{x}, x) = 0$$

om den får en rutin som beräknar

$$\Delta = g(t, \dot{x}, x)$$

när t , \dot{x} och x är kända.

ODE-lösare löser

$$\dot{x} = f(t, x)$$

om de får en rutin som beräknar $f(t, x)$ när t och x är kända.

Generera ekvationerna

1. Instantiera modellen
 - a. härled ospec terminalkomponenter
 - b. skapa unika objekt för alla modellkomponenter
2. Ekvationerna till intern form
 - a. lexikalisk- och syntaxanalys
 - b. lös upp variabelreferenser
 - c. typkonsistens
3. Härled kopplingsekvationerna

Problemet har nu formen

$$g(t, x) = 0$$

t är tiden

x är vektor av okända variabler som kan förekomma oderiverade eller deriverade en eller flera gånger

Analysera och manipulera

Testa konsistens och underlätta numerisk lösning.

1. Strukturellt singulärt?
2. Separera bort tidskonstanta variabler
3. Analysera och manipulera index
4. "Eliminera" algebraiska variabler
5. Sortera i dynamiska och ut-variabler

BLT-partitionering

"Output set":

1. Varje ekvation associeras med en variabel. Den måste uppträda i ekvationen.
2. En variabel är bara associerad till en ekvation

Partitionering:

Permutering av variabler och ekvationer

BLT-partitionering:

- Nedre-block-triangulär form
- Effektiva algoritmer finns och ger minimala diagonalblock

Strukturellt singulärt?

Problemet $g(t, x) = 0$ är strukturellt singulärt om det inte finns något output-set.

Vi skiljer inte på x och dess derivator.

Detekterar inte om en ekvation förekommer flera gånger.

Tidskonstanta variabler?

1. BLT-partitionera $g(t, x) = 0$
Skilj inte på x och dess derivator
2. För varje block om
 - a. t inte förekommer
 - b. blockets variabler inte förekommer deriverade i blocket
 - c. övriga variabler som förekommer i blockets ekvationer är tidskonstanta så är blockets variabler tidskonstanta.

DAE-index

Om $g(t, \dot{x}, x, v) = 0$ kan transformeras algebraiskt till

$$\dot{x} = f(t, x, p, c)$$

$$v = V(t, x, p, c)$$

så är DAE-index < 2 .

Annars finns det algebraiska relationer mellan dynamiska variabler.

Problemet är svårt för DAE-lösare:

1. Derivera de algebraiska relationerna; orsakar problem med numerisk drift
2. Eliminera dynamiska variabler.

Manipulering av DAE-index

1. BLT-partitionera $g(t, x) = 0$ med högsta derivatan som okänd
2. Om partitioneringen lyckas så är struktur-DAE-index < 2
3. Pantelides algoritmen pekar ut minimala system innehållande algebraiska relationer mellan deriverade variabler
 - a. försök eliminera en dynamisk variabel
 - b. om det inte går: derivera?

"Eliminera" algebraiska variabler

Betrakta numerisk lösning av $h(v) = 0$
Behöver en residualrutin för $\Delta = h(v)$.

Exempel: Strukturmatrisen

	a	b	c	d	e	f
1	x	x				
2	x	x				
3	x	x	1			
4	x		x	x		x
5				x	x	x
6	x				x	x

Kan beräkna c internt;
Ingen substitution!

För vårt problem:

1. BLT-partitionera $g(t, \dot{x}, x, v) = 0$ med x som känd.
2. Undersök vilka v_i som kan "elimineras"
 - a. symbolisk manipulering
 - b. tearing

Sortera bort rena "utvariabler"

För att beräkna de deriverade variablerna behöver man inte beräkna alla variabler:

$$0 = f(t, \dot{x}, x, v)$$

$$0 = h(t, \dot{x}, x, v, y)$$

Lätt att bestämma genom att analysera BLT-partitioneringen nedifrån.

Terminalerna till strukturerade realiseringar inför extra variabler som lätt kan elimineras $a = \text{konstant} * b$ och sorteras till utvariabler.

Kodgenerering för simulering

ODE- och DAE-lösare

1. använder vektorer
2. bara första derivator

Generera kod för

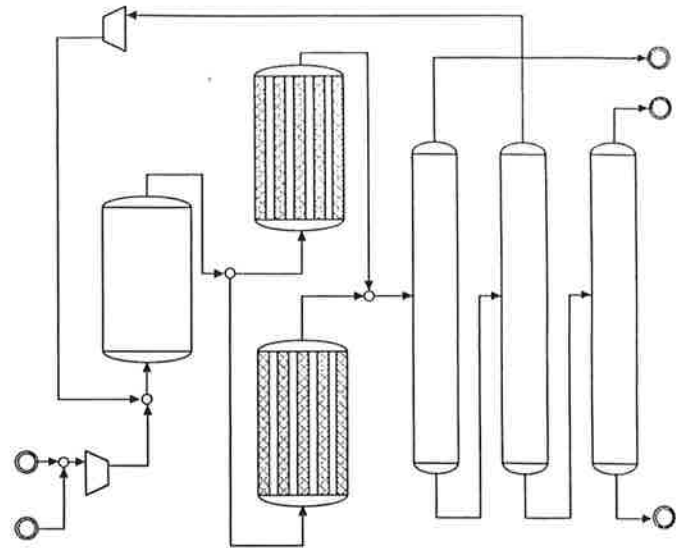
1. tidsinvarianta variablerna
2. dynamiska variablerna
(+ de "eliminerade" variablerna)
 - a. om ODE-lösare önskas och möjligt generera rutin för derivatorna
 - b. annars residualrutin
3. utvariablerna

Strukturerad Modellering av Kemiska Processer

Ett Objektorienterat Synsätt

Bernt Nilsson

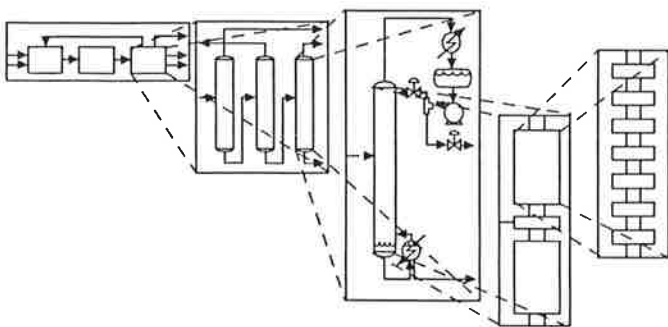
Processmodell



Objektorienterad modellering underlättar:

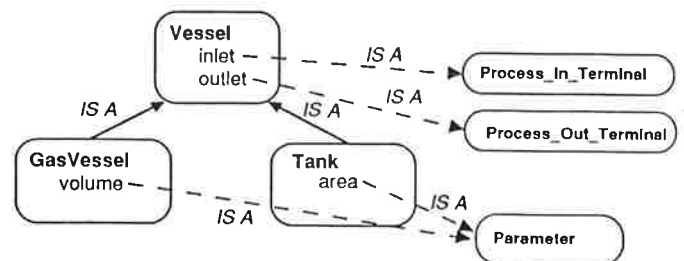
- Modellutveckling
- Modellåteranvändning

Process-struktur-dekomponering



- Hierarki ⇒ Abstraktion
- Koppling/Terminal ⇒ Särkoppling
- Ärvning ⇒ Återanvändning

Submodell-beskrivning



- Intern struktur ⇒ Utveckling
- Ärvning ⇒ Återanvändning

Generiska Modeller I

```
Vessel IS A Model WITH
terminals:
  Inlet IS A Process_In_Pipe;
  Outlet IS A Process_Out_Pipe;
parameters:
  No_Comp IS A Parameter WITH value:=2 END;
  Phase IS A Parameter;
constraints:
  Inlet.Composition.Lenght :-
    Outlet.Composition.Lenght :- No_Comp;
  Inlet.Phase :- Outlet.Phase :- Phase;
END;
```

```
Process_In_Pipe IS A RecordTerminal WITH
components:
  Pressure IS A Pressure_Terminal;
  Flow IS A Flow_In_Terminal;
  Temperature IS A Temperature_Terminal;
  Composition IS A VectorTerminal WITH
    Lenght := 1;
  CompType IS A Concentration_Terminal;
END;
Phase IS A SimpleTerminal WITH
  variability:=constant;
END;
END;
```

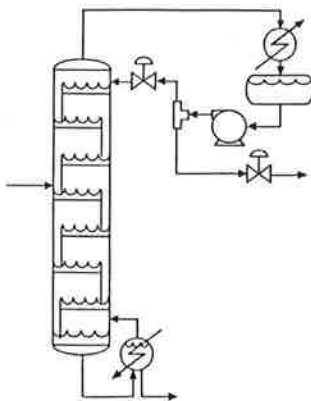
- Parameterisering ⇒ Generiska modeller

Generiska Modeller II

```
GasVessel IS A Vessel WITH
parameters:
  Volume, GasConst, HeatCap IS A Parameter;
  MoleWeight TYPE Row[No_Comp];
  Phase := "Vapour";
realizations:
  Behaviour IS A Primitive WITH
  variables:
    composition IS A Column[No_Comp];
    pressure, temperature, density, energy IS A Real;
  equations:
    pressure*Volume =
      GasConst*SUM(composition)*temperature
    Volume*DOT(composition) =
      Inlet.Flow*Inlet.Composition -
      Outlet.Flow*Outlet.Composition;
    density = MoleWeight*composition;
    DOT(energy) =
      Inlet.Flow*density*HeatCap*Inlet.Temperature -
      Outlet.Flow*density*HeatCap*Outlet.Temperature;
    energy = density*Volume*HeatCap*temperature;
    Inlet.Pressure = pressure;
    Outlet.Pressure = pressure;
    Outlet.Temperature = temperature;
    Outlet.Composition = composition;
  END;
END;
```

- Parameterisering ⇒ Generiska modeller

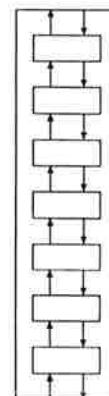
Destillationskolonn



Särkoppling:

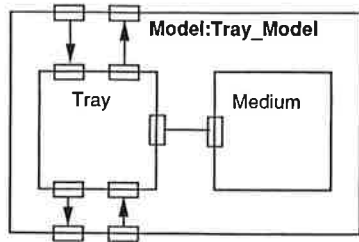
- Konfiguration av enheten
- Design av kolonnen
- Maskinmodell
- Mediamodell

Reguljära strukturer



- Parameterisering ⇒ Generisk struktur (finns ej i SEE-prototypen)

Media/Maskin-dekomponering



- Dekomponering ⇒ Generiska modelldelar
- Parameterisering ⇒ Generisk struktur

"Modelleringsmetodik"

Språket ger Utvecklaren:

- Strukturingskoncept
- Ärvningsmekanismer
- Enkel parameterisering

Utvecklaren ger Användaren:

- Dekomponerande modeller
- Avancerad parameterisering

Slutsatser

Återanvändning och utveckling av modeller underlättas avsevärt.

Modeller för kontinuerliga kemiska processer för fluider kan utnyttja denna typ av objektorienterad modellering med stor framgång.

Det finns behov av några tillägg till Omola för att passa processtillämpningar perfekt.

CACE-projektet

1. Bakgrund och mål
2. Resultat
3. Erfarenheter

Bakgrund

Dagens CACE-verktyg är användbara
utför numeriska räkningar
och ritar grafer

men användarna önskar

- bättre användargränssnitt
- stöd för tankeprocessen
- integrerad miljö
från specifikation till drift av en process.
- utvidgbarhet

Hypotes vid starten

Dagens verktyg designades för 10 – 20 år
sedan.

Den enorma utvecklingen inom datalogin

- Arbetsstationer
- Objekt-orienterad programmering
- Databaser
- Grafik och användargränssnitt
- Expertsystemteknik
- Datoralgebra

borde tillåta bättre CACE-verktyg.

Mål

- Undersöka hur de nya landvinningarna
inom datalogin kan förbättra CACE-
verktygen.
- Utveckla några prototypsystem.
- Etablera internationella kontakter.

Resultat

1. Hibliz – grafikbaserat MMI för simulering
2. Expertsystemschnitt för identifieringspaket
3. Formelbehandling av flervariabla linjära system
4. Expertreglering
5. Verktyg för modellering och simulering
 - Designförslag på kärna
 - Prototypimplementering i KEE
 - Tillämpningsprojekt

Modeller är centrala

Matematiska modeller och simulering är viktigt i alla ingenjörssammanhang

- Analys, konstruktion, validering
- Operatörsutbildning, -träning
- Driftsplanering, operatörsstöd
- Fel detektering, -analys

Modellering och simulering är intimt förknippade med varandra.

Modellrepresentation

Svårt och tidskrävande att utveckla modeller.
CACE-verktyg behövs:

- Integrerade verktyg
 - a. Gemensam representation
 - b. Modelldatabas – Modellbibliotek
- Återanvändning
 - a. Olika beräkningar
 - b. Olika system
- Användargränssnitt
 - a. Rätt abstraktion
 - b. Grafik

Hur?

- Ekvationsbaserade modeller
- Strukturerade modeller

Ekvationsbaserade modeller

Modellerna skall beskriva egenskaper och relationer:

inte en beräkningsrutin
VAD, inte hur

Stöder åter användning för olika beräkningar;
Automatiskt generering av:

- effektiv simuleringskod
- kod för stationära punkter
- linjära beskrivningar
- beskrivningar till existerande paket
- reglerprogram
- layout för specialtillverkade analoga eller digitala VLSI-kretsar som implementerar regulatorn.

Andra fördelar:

- Modellutvecklarens naturliga språk
- Bra dokumentationsform
- Konsistenstester

Strukturerade modeller

Modellstrukturering

- Hierarkisk modelldekomposition
- Modelltyper
- Klasshierarki – Ärvning
- Multipla beteendebeskrivningar
- Parameterisering

Fördelar:

- Nödvärdigt för stora system
- Återanvändning av modelldelar
- Underlättar dokumentation
- Möjlighet att modifiera eller byta ut delar

Publikationer

- 1 doktorsavhandling
 - Karl-Erik Årzén
- 3 licentiatavhandlingar
 - Jan Erik Larsson
 - Per Persson
 - Bernt Nilsson
 - Mats Andersson
- 8 examensarbeten

- 5 artiklar
- 23 konferensbidrag

- 8 slutrapporter till STU
- 2 STU-seminarier

Internationella kontakter

5 gästforskare

Den internationella kontaktnätet är stort.

Några exempel:

DK DTH
 NL Delft, Eindhoven
 F INRIA
 GB SERC, UMIST, Cambridge, Swansea,
 CEGB, Imperial College
 USA Maryland, RPI, LLNL, Berkeley,
 Santa Barbara, GE, Mathworks,
 AT&T Bell Labs

Objekt-orienterad programmering

Strukturering och återanvändning:

1. Abstrakta datatyper
2. Ärvning

Naturlig abstraktion för reglertekniker:

- Blockdiagram, väldefinierade snitt.
- In/utsignal beskrivning – Intern modell

Inte självklart hur idéerna bör användas:

1. För implementering
 - a. integrerad miljö
 - b. användaranpassade gränssnitt
 - c. utvidgbarhet
2. För modellutveckling
 - a. modelldekomponering
 - b. återanvändning
 - c. konsistens
 - d. dokumentation och indexering

Databaser

Databasen är central

- modellbibliotek
- parameterdata
- mätdata
- beräkningsresultat

används av alla verktyg.

Gemensam representation för integrering.

Dagens databaser kan effektivt hantera en stort mängd, oberoende data.

I CACE är mängden liten, men relationerna är komplexa.

Objekt-orientering är en lovande ansats.

Grafik och användargränssnitt

Direkt manipulering

- konkreta representationer
- välj, peka, undersök och utför
- strukturinformation

Det behövs användaranpassade gränssnitt:

- olika användarkrav och -behov
- olika hårdvara
- olika interaktionsstilar

Arbetsamt att implementera användarsnitt:

- Portabilitet
Grafik- och fönsterstandard på väg
Användaren vill ha enhetlighet på sin dator
- Stödprogramvara
Idag primitiv
Kommersiella UIMS på väg

Expertsystemteknik

Programmeringstekniska idéer:

- Deklarativ programmering
- Representation av kunskap
- Kvalitativt resonemang
- Påskyndat och påverkat arbetsstationer, grafik, objekt-orientering

Sätt att inkludera och utnyttja kunskap om

1. Tillämpningen
2. Verktygen
3. Resultaten

Svårt att samla in kunskap

- Systematisering
- Öppna frågor

Icke-monoton logik, realtidskrav

Datoralgebra

Användning

- Analytiska uttryck kan ge bättre insikt
- Underlätta numerisk lösning
- Generering av kod för numerisk lösning

Kommersiella system

- MACSYMA, REDUCE, Scratchpad, Maple, muMATH, Mathematica
- "Bättre än 20-poängs-matematikstudent"
- Interaktiva paket, svåra att integrera som verktyg

Modellering av Sockerkristallisation

DUP-projekt (ansökan)

Björn Wittenmark, Reglerteknik
Claes Gudmundsson, Sockerbolaget

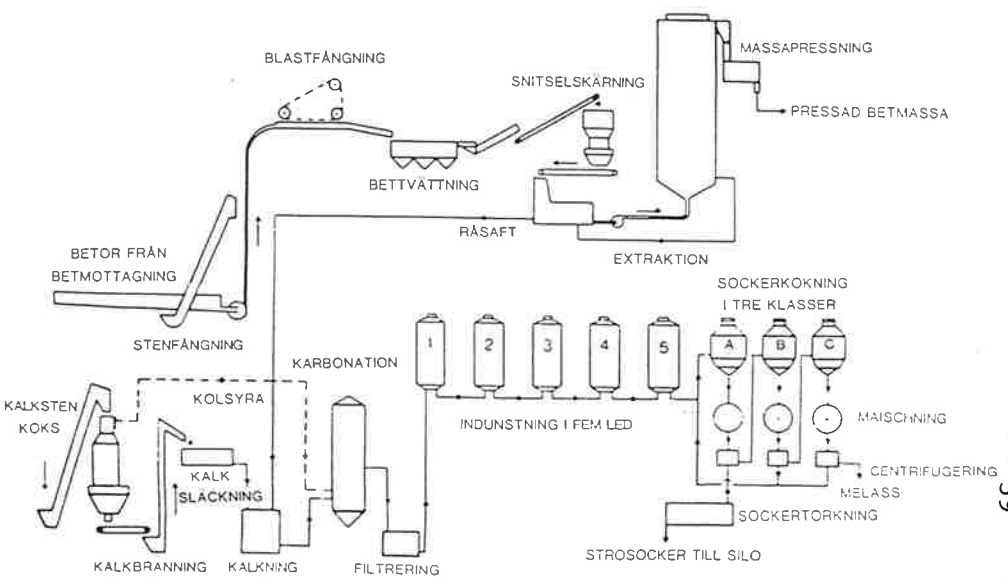
- Modellbygge
- Utvärdering av modellen
- Styrstrategier
- Driftsanpassning

Projektfaser

- Följande egenskaper skall underlättas vid modellbygge:
 - Modellutveckling
 - Modellåteranvändning
 - Modellförfining (utvärdering)
 - Modellunderhåll

Modelleringsmetodik

SOCKERTILLVERKNING



P R O T O K O L L

från telefonmöte med STUs styrgrupp för ramprogram CACE den 24 februari 1989 kl 09 00 - 10 00.

Deltagare: Sven Gunnar Edlund
 Karl Eklund
 Sven Erik Mattsson
 Gustaf Söderlind
 Arne Otteblad
 Karl Johan Åström

- § 1 Formalia
 Ordförande: Sven Gunnar Edlund
 Sekreterare: Arne Otteblad
 Justeringsman: Sven Erik Mattsson
- § 2 Det utsända protokollet från styrgruppsmötet i Lund den 23 november 1988 godkändes.
- § 3 Sven Erik Mattsson hade inför mötet skickat ut en skrivelse angående den tidigare beslutade prototypimplementeringen i projektet (bilaga 1). På styrgruppens önskan hade Sven Erik M vidare skrivit ett förslag till fortsättning av verktygsutvecklingen efter CACE-programmets avslutning (bilaga 2). Formellt avslutas CACE den 30 juni 1989 men enligt Arne Otteblad kan STU förmodligen stötta verksamheten med 750 - 1000 kkr under budgetåret 1989/90. Styrgruppen instämde helt enligt de förslag som framförts i bilagorna och Sven Erik M ombads skriva en ansökan till STU, som kunde diskuteras vid ett nytt telefonmöte i april.
- § 4 Efter Arne O:s önskemål hade Sven Erik M skrivit ner ett antal motiveringar för behovet av projektstöd inom STUs programområde "Informationsteknologins tillämpningar". Styrgruppen instämde i argumenten och tillstyrkte med eftertryck behovet.
- § 5 Styrgruppen diskuterade höstens planerade konferens och föreslog v. 43 som lämplig tidpunkt. Sven Erik M och Arne O skulle undersöka detta vidare. Sven Erik M åtog sig att till nästa möte skriva ett förslag till konferensprogram.
- § 6 Nästa möte avhålls som tidigare har bestämts den 6 april 1989 kl 09 00 som telefonmöte.

Vid protokollet



Arne Otteblad
 STU

Justeras



Sven Erik Mattsson
 LTH

Om prototypimplementeringen i projektet "Verktyg för modellutveckling och simulering"

Sven Erik Mattsson

Inledning

Vid styrgruppsmötet för ramprogram CACE den 23/11 1988 diskuterades olika alternativ till fortsättning och sätt att sprida resultaten från projektet. Ett alternativ som diskuterades var att utveckla en basprototyp som någon sedan kan utveckla till en kommersiell produkt.

Idag implementeras en prototyp med hjälp av KEE och Common Lisp. Fördelen med detta har varit att vi kunnat utveckla prototyper med en liten insats av programmerare. Nackdelen är att KEE och Common Lisp fordrar kraftfulla datorsystem med ett primärminne på minst 8 Mbyte och helst 16 Mbyte. Vidare är KEE en dyr programvara. Detta gör att den prototyp som nu utvecklas i projektet inte kan förväntas få någon större spridning. Om man vill nå en större spridning bör man alltså implementera en prototyp med billigare och mer spridda språk och mjukvarukomponenter. Ett möjligt implementeringsspråk som har nämnts i diskussionen är C++.

Hur påverkas dagens arbete av eventuella planer på att göra en implementering av en prototyp utan hjälp av KEE och Common Lisp i ett senare projekt? Skall vi avbryta implementeringen i KEE och Common Lisp? Denna fråga ställde jag explicit till styrgruppen under mötet den 23/11 1988 och fick ett klart svar att vi skulle fortsätta med implementeringen i KEE och Common Lisp. Jag har sedan efter mötet blivit ombedd att skriftligt framföra mina synpunkter på detta svar inför styrgruppsmötet den 24/2 1989.

Synpunkter

Vi i CACE-projektet anser att implementeringen av en prototyp i KEE och Common Lisp skall fortsätta som planerat.

Motiveringen är delad i två steg. Först motiveras varför man bör göra en prototypimplementering i projektet. Sedan diskuteras val av implementeringsverktyg.

Avsikten med en prototypimplementering är

1. att få en prototyp för demonstrationer som kan
 - a. illustrera idéer och begrepp mera konkret än verbala beskrivningar
 - b. ge visibilitet och väcka intresse
 - c. indikera att våra idéer är genomförbara
2. att få en prototyp som kan ge oss återkoppling
 - a. från demonstrationer
 - b. vad som är realistiskt att implementera
 - c. hur implementeringen skall göras

Alla dessa skäl är viktiga. Under implementeringsarbetet har vi många gånger speciellt lagt märke till punkterna 2b och 2c. Vid implementering upptäcker man ofullständigheter och oklarheter i specifikationen. Vidare finner man att en del saker är mycket svårare att implementera än man trott, men mindre ändringar eller förenklingar gör att svåra implementeringsproblem undviks. Implementeringsarbetet är alltså en viktig del i designarbetet.

När det gäller val av implementeringsspråk kan man givetvis inte byta det så här i slutet av projektet. Men låt oss trots detta nämna vilka skillnaderna mot att implementera i till exempel C++ är.

Den grundläggande programstrukturen och den interna representationen kan behållas oförändrad. En viktig skillnad är dock att C++ som är ett kompilerat språk inte tillåter att man skapar nya klasstyper dynamiskt under exekveringens gång, utan alla klasstyper måste vara definierade vid kompilering. Vårt modelltypbegrepp måste tillåta att användaren skapar nya modelltyper interaktivt. Eftersom KEE tillåter att man definierar nya klasstyper dynamisk har vi kunnat utnyttja dess klassbegrepp och ärvningsmekanismer för att direkt avbilda vårt modelltypbegrepp. Vid implementering av vårt modelltypkoncept i C++ måste man implementera ett mellanlager för att hantera definition av nya modelltyper och ärvning mellan modelltyper.

En annan mycket viktig skillnad är att KEE tillhandahåller en rad grafiska rutiner och rutiner för att definiera användargränssnitt. Detta har gjort att vi har kunnat hålla arbetet med att implementera ett användargränssnitt på ett minimum. Vid implementering i C++ kommer implementeringen av grafikrutiner och användargränssnitt att uppta en mycket stor del av arbetet. Det finns en del grafikpaket för C++ som till exempel InterViews, men de är långt ifrån så kraftfulla som KEEs.

Sammanfattning

Sammanfattningsvis anser vi i CACE-projektet som det framgår av ovan givna skäl att det är helt nödvändigt bedriva prototypimplementeringsverksamhet som planerat i vårt projekt.

Implementeringen i KEE och Common Lisp kan utgöra en bra grund för en implementering i ett annat språk såsom C++. Man kan behålla den grundläggande programstrukturen och den interna representationen av data och modeller, men man måste bygga en del mellanlager. Vidare kommer en stor del av arbetet att upptas av att implementera grafik och användarsnitt. De flesta användare har här stora krav efter bekantskap med Macintosh. Man måste om och när det blir aktuellt noga undersöka möjligheterna att få och använda färdiga komponenter. Utvecklingen på detta område är snabb och vi följer den noga.

STYRELSEN FÖR TEKNISK UTVECKLING	
Ank. 1989 -02- 20	
D/Dnr	
Aa	Sign

Bilagga 42

Förslag till fortsättningar efter CACE-projektet

Sven Erik Mattsson

Inledning

Vid styrgruppsmötet för ramprogram CACE den 23/11 1988 diskuterades olika alternativ till fortsättning och sätt att sprida resultaten från projektet. Nedan ges först en sammanfattning av tänkbara alternativ och sedan förslag till fortsättningar inför styrgruppsmötet den 24/2 1989.

Alternativ

Det finns flera naturliga fortsättningar av projektet:

1. grundforskning,
2. tillämpningsstudier och
3. implementeringsprojekt.

Grundforskning

Under projektet har vi funnit flera fundamentala problem som är mycket intressanta. Ett sådant är hur man skall hantera diskreta händelser i samband med kontinuerlig simulering. En ansökan för ett sådant projekt "Diskreta händelser i kontinuerliga modeller (Discrete-event elements in continuous modeling)" har sänts till STUFs ansökningsomgång den 20 januari 1989. Det är svårt att säga något om möjligheterna, men det lär ha inkommit ansökningar för 4-6 gånger av disponibelt belopp.

Ett annat intressant alternativ är presentationsformer för reglersystem och dess struktur. Detta ställer andra krav än presentation av processer och dess struktur. Växelverkan mellan process och reglersystem behöver inte vara rumsligt lokal på samma sätt som växelverkan mellan processkomponenter. Ett reglersystem kan ha många signaler som kan komma från helt skilda delar av en process. Ett blockdiagram som innefattar reglersystem kan därför lätt bli rörigt.

Tillämpningsstudier

De resultat vi erhållit kan tillämpas på många olika sätt. Tillämpningarna bör lämpligen utföras tillsammans med externa parter. Målen med dem kan vara flera såsom att

1. sprida resultaten,
2. lägga grund för en bred användning,
3. få en återkoppling från användare,
4. lägga grunden till allmänna modellbibliotek och
5. utveckla tillämpningsspecifika verktyg.

Som möjliga kandidater för applikationsstudier i ett tidigt skede ser vi

1. *Kemi- och massa-pappers-branscherna.*
 - a. DUP-ramprogrammet har skapat ett intresse inom denna bransch för modellutveckling och kontakter har knutits. Det är naturligt att inom DUP-ramprogrammet formulera och finansiera tillämpningsstudier.
 - b. Modellering av kemiska processer studeras i CACE-projektet.
 - c. Vi har forskare som är kunniga i och intresserade av modellering av kemiska processer.
 - d. IT-4 projektet "Knowledge-based real-time control systems" kommer för sin demonstrator att använda en steriliseringsprocess.
 - e. Vi har goda kontakter med utländska grupper som sysslar med modellering av kemiska processer såsom Tyréus, DuPont; Asbjornsen och McAvoy, University of Maryland; Sargent, Imperial College.
2. *Kraftindustrin.*
 - a. Sydskraft har visat intresse för våra idéer.
 - b. Det finns god kompetens i modellering av kraftverk och kraftsystem vid institutionen.
 - c. Institutionen har gästprofessur i reglering av kraftsystem.
 - d. Vi har goda kontakter med många utländska grupper inom området såsom Rees, Australien och Maffezzoni, Milano.
 - e. Det finns många bra modeller för kraftverk tillgängliga.

Finansiering har indikerats ovan i de olika alternativen. En annan möjlighet efter 1 juli 1990 kan vara fria projekt inom området för modellutveckling och simulering under STUs nya program Informationsteknologins tillämpningar.

Ett grundläggande problem med tillämpningsstudierna är att vi behöver tillgång till verktyg. Prototypimplementeringen i KEE och Common Lisp kan användas i början, men sedan måste vi ha en prototyp som kan göras mer tillgänglig.

Implementeringsprojekt

För att kunna sprida resultaten från CACE-projektet i tillämpningsprojekt behöver vi tillgång till implementeringar av verktygen. I CACE-projektet implementeras idag en prototyp med hjälp av KEE och Common Lisp. Fördelen med detta har varit att vi kunnat utveckla prototyper med en liten insats av programmerare. Nackdelen är att KEE och Common Lisp fordrar kraftfulla datorsystem med ett primärminne på minst 8 Mbyte och helst 16 Mbyte. Vidare är KEE en dyr programvara. Detta gör att den prototyp som nu utvecklas i projektet inte kan förväntas att få någon större spridning. Om man vill nå en större spridning bör man alltså implementera en prototyp med billigare och mer spridda språk och mjukvarukomponenter.

Vi anser det inte vara en högskolas uppgift att utveckla, marknadsföra och underhålla kommersiella och professionella programvaror. Men vi inser att vi bör ta vårt ansvar här och att vi kan få nyttiga erfarenheter från projekt av begränsad omfattning. Erfarenheter från Simmon visar att användbara programkomponenter är ett mycket bra sätt att föra ut resultat. Ett förslag till att utveckla en basprototyp som någon sedan kan utveckla vidare till en kommersiell produkt diskuterades på styrgruppsmötet den 23 november 1988.

Förslag

De ovan diskuterade alternativen är inte uteslutande utan kompletterar varandra på olika sätt och är av olika intresse för de som idag arbetar i CACE-projektet. Vårt förslag är att de 750 - 1000 kSEK som finns tillgängliga för fortsättning av CACE-projektet under 89/90 används till ett implementeringsprojekt. Resurser för

tillämpningsprojekt får sedan sökas på andra håll till exempel inom DUP-ramprogrammet. Vi är också intresserade av att utvidga våra begrepp till modeller med diskreta händelser som till exempel modeller av batchprocesser. Vi kommer säkert att stöta på detta krav i tillämpningsprojekten. En ansökan om ett projekt med denna inriktning har sänts till STUF.

Det är viktigt att prioritera ett implementeringsprojekt. Användbara programkomponenter är ett mycket bra sätt att föra ut resultat. Det gör det möjligt för intresserade att själva få prova och det ger oss återkoppling. För det andra, behöver vi tillgång till användbara verktyg i tillämpningsprojekten. Ett implementeringsprojekt är alltså en direkt fortsättning på forskningen för att föra ut resultaten och det bör ligga tidigt innan tillämpningsprojekten avancerat för långt. Vi inser att de ovan nämnda resurserna inte är stora, men de ger oss nu en tidig möjlighet att implementera användbara komponenter. För att hålla projektet inom de ovan angivna ramarna är vårt förslag att arbetet skall koncentreras på att implementera en kärna för modellutveckling och simulering. Med en kärna menar vi rutiner för att handha och manipulera den interna representationen. I vår design finns det en klar separation mellan användargränssnitt, intern representation av data och modeller och bearbetningsrutiner. Avsikten är att undvika att lägga ner mycket arbete på användargränssnittet. I ett projekt att utveckla en kommersiell produkt kommer implementeringen av användargränssnitt att vara en mycket tung del. Det finns goda skäl att inte nu lägga ner mycket arbete på användargränssnitt:

1. Vi kan illustrera våra idéer och få användbara verktyg till tillämpningsprojekten med ett enkelt användargränssnitt.
2. Det finns flera möjliga utformningar av ett bra användargränssnitt. Tillämpningsprojekten kan ge oss erfarenheter. Det är naturligt att göra arbeten på användargränssnitt i ett tillämpningsprojekt, liksom att tillverka nya verktyg.
3. Det finns idag mycket få komponenter för att definiera och implementera användargränssnitt, men utvecklingen inom området är snabbt. Man kan förvänta sig att det inom 1 - 2 år finns fler intressanta komponenter tillgängliga.

Det är alltid svårt att bedöma resursåtgången i ett programmeringsprojekt, men vi bedömer det som möjligt att inom ovan nämnda ram implementera en kärna för modellutveckling och simulering för modeller som beskrivs av differential-algebraiska ekvationer och om det är möjligt också differensekvationer. Verktygen bör också kunna handha vektor- och matrisuttryck och enklare former av datoralgebra.

Som basimplementeringsspråk avser vi att välja C++. Det ger oss en bra portabilitet. Den grundläggande programstrukturen och den interna representationen från prototypimplementeringen med KEE och Common Lisp kan till största delen bibehållas. Det finns ett grafikpaket i C++ som heter InterViews, som har utvecklats på Stanford och är allmänt tillgängligt. Det är inte så väl integrerat som KEEs grafik, men det har rutiner för menyhantering, dialogrutor och textredigering.

Vi har goda kontakter med den grupp på AT&T Bell Laboratories som ansvarar för utvecklingen av C++. I denna grupp ingår Bjarne Stroustrup som har skapat C++. Speciellt god kontakt har vi med Andrew Koenig, som är en av de huvudansvariga för version 2 av C++. Han har varit på besök hos oss och är intresserad av vårt projekt. Andrew håller på att skriva en bok om C++ och är intresserad av att hålla en seminarierie om C++. En person med Andrews erfarenheter av programmering i C++ skulle vara en stor tillgång i början av projektet för att hjälpa oss att få en riktig grundarkitektur på kärnan. I preliminära diskussioner har han uttalat sig intresserad av att gästforsa hos oss en till två månader. Vi skulle bara behöva stå för rese- och bostadskostnader.

Motiv för fria projekt inom området modellutveckling och simulering i STUs planerade program Informationsteknologins tillämpningar

Sven Erik Mattsson

Stycket nedan argumenterar för fria projekt inom området modellutveckling och simulering i STUs programskrivning för "Informationsteknologins tillämpningar".

Fria projekt inom området modellutveckling och simulering

Den hårdnande konkurrensen kräver både bättre och billiga produkter. Energiförbrukningen måste minskas och miljön får inte förstöras. Industrin är därför allt mer intresserad av att konstruera automatiserade produktionsprocesser som utnyttjar energi och råmaterial på bästa sätt och samtidigt är miljövänliga.

Informationsteknologin har gett oss kraftfulla hjälpmedel som möjliggör omfattande konstruktionsberäkningar. Därigenom kan man på konstruktionsstadiet undersöka många olika processkonfigurationer och utföra optimeringar. Krav på miljövänlighet och låg energiförbrukning har lett till slutna processer med många återcirkulationer. Sådana processer kräver sofistikerad styrning. Matematiska modeller av industriella processer är nödvändiga för att konstruera processerna och deras styrsystem. Modeller och simuleringsverktyg behövs också i simulatorer för träning eller som beslutstöd för operatörer. Simulatorer och avancerade matematiska metoder för analys och design har hittills huvudsakligen använts där det varit omöjligt att prova sig fram till en någorlunda acceptabel lösning eller för att myndigheter har krävt det av säkerhetsskäl. Typiska exempel är elkraftproduktion, flyg- och rymdteknik.

Det finns ett behov av att överföra kunskap om nya metoder och datorverktyg för modellutveckling och simulering till en rad intressenter inom högskola och industri. Inom till exempel CACE-ramprogrammet har sådan kunskap byggts upp. Metoder har utvecklats för att underlätta modellutveckling och återanvändning av modeller till olika ändamål och för liknande processer.

Fria projekt med användare och utvecklare av datorstödda verktyg för modellering och simulering är till gagn för alla. I projekt med användare både inom högskola och industri som har processkunskap och någon erfarenhet av modellering och simulering får användarna lära sig nya metoder och tekniker. Utvecklarna av modelleringsverktygen får återkoppling och kan modifiera och förbättra verktygen. För speciella tillämpningar kan de tillverka speciella verktyg eller specialanpassa användargränssnitten och modellbibliotek kan byggas som kan vara till nytta inte bara för den deltagande parten utan en hel bransch. Denna tekniköverföring är speciellt viktig eftersom det finns industrigrupper där man nu inte har någon större erfarenhet och kunskap i modellutveckling och simulering. Detta kan också leda till att en de facto standard för att beskriva modeller kan etableras vilket skulle drastiskt bidra till tekniköverföringen.

P R O T O K O L L

Från telefonmöte med STUs styrgrupp för ramprogram CACE den 6 april 1989 kl 09 00 - 10 00.

Deltagare: Sven Gunnar Edlund
 Karl Eklund
 Sven Erik Mattsson
 Gustaf Söderlind
 Arne Otteblad

- § 1 Formalia
 Ordförande: Sven Gunnar Edlund
 Sekreterare: Arne Otteblad
 Justeringsman: Sven Erik Mattsson
- § 2 Styrgruppen hade inför mötet fått kopior av STU-ansökningen för ett implementeringsprojekt under 1989/90. Samtliga ansåg att projektet och ansökan var bra och Styrgruppen rekommenderade STU att stödja projektet i äskad omfattning.
- § 3 Sven Erik Mattsson hade utarbetat ett förslag till höstens CACE-seminarium.

 En del förslag till förbättringar framfördes och Sven Erik M arbetar vidare med programmet.

 Datum för seminariet bestämdes till den 24 oktober. Seminariet bör om möjligt aviseras i IT-aktuellt och/eller Teknik i Tiden.
- § 4 Nästa möte bestämdes till 12 juni i Lund. (Ändrades senare till telefonmöte den 13 juni)

Vid protokollet



Arne Otteblad
 STU

Justeras



Sven Erik Mattsson
 LTH

**Kernel for
Model Development and Simulation**

A project proposal
to STU
1989-03-30

from
Sven Erik Mattsson

Department of Automatic Control
Lund Institute of Technology
Box 118, 221 00 LUND

1. Introduction

Significant interest in modeling and simulation is manifested in the DUP program and from industry. The underlying aims are to design and run plants more efficiently and to create a better environments for the operators. Today's CAE tools for model development and simulation are far from satisfactory. Most of today's continuous time modeling languages follows the basic structure of the CSSL definition (Strauss, 1967), which has had a profound impact. It has served well for over 20 years, but the development of computing science and computer hardware has been enormous during this time.

In the end of 1984 a 5 years' research programme "Computer Aided Control Engineering, CACE" financed by STU was started to investigate the possibilities to capitalize on the development of computing science and computer hardware. During the two last years (July 1987 – June 1989) the problem of supporting model development and simulation are addressed in a project "Tools for model development and simulation" (STU project 87-2503 and 712-88-2508). The major result of this project will be a design proposal containing concepts and tools to support model development and simulation. Overviews of basic ideas are given in Mattsson (1988, 1989) and Andersson (1989). The design includes also an internal representation to support these concepts (Andersson, 1989). Experience from the prototype indicate that the ideas are sound and the prototype may indeed serve as the basis for a new generation of simulators. The prototype implementation is written in Common Lisp and KEE. The advantage of KEE is it could be implemented with a modest effort. However, since KEE is expensive, we do not expect the prototype to be widespread. To make our tools generally available, it is necessary to implement them using cheaper and more commonly available languages and software components.

We think that it is not the task of a university to develop, market and maintain commercial and professional software. But we realize that we have a responsibility of spreading the results of our project and making them generally available. The experiences from Simnon show that useful program components are a very good way of spreading new ideas and methods. Consequently, this application contains a project proposal to implement a kernel for model development and simulation, which someone else can develop further to a commercial product.

Another good way of spreading new ideas and methods is to have joint application projects with developers and users. The developers also get feedback and can modify and improve the tools. For a special application they can develop customized tools and user interfaces. Model libraries can be built which can be of use not only for the participating part, but also for a whole branch. However, these application projects assume that there are useful tools available. We would therefore at this stage give priority to an implementation project.

Before giving a specification of the kernel to be implemented and a project plan, it is necessary to give a background by outlining some basic ideas to put the prototype in a context and to make the proposal more concrete.

2. Basic ideas

The basic aim of the project proposed is to implement a useful kernel for model development and simulation along those specifications developed in the CACE project. By a kernel we mean the routines to manipulate the internal representation. In our design there is a clear separation between the user interface, internal representation of data and models and processing tools. This separation makes the design more

flexible and allows customized user interfaces. In a project to develop a commercial product, the implementation of user interfaces will be a major part. Our approach is to avoid to put too much effort on implementing user interfaces in the project proposed. There are several good reasons:

1. It is desirable to get useful tools as soon as possible.
2. We can illustrate our ideas and get useful tools with simple user interfaces.
3. There are several possible ways of designing a good user interface. The application projects can give us experiences. It is natural to develop customized user interfaces in the application projects, as well as develop new special tools.
4. Today there is very few high-level software tools to define and implement user interfaces, but the development within the area is fast. In 1 – 2 years there may be several useful components available.

Our approach to make a kernel relies on the belief that various users could agree upon a small, basic, common set of concepts; A collection of basic objects like *models* and *terminals* with specified properties and operations. The system Mathematica (Wolfram, 1988) for numeric and symbolic calculations is built in the same way. It has a kernel, which does the computations and a user interface which manages the interaction with the user. The kernel works in the same way on all computers that run Mathematica, but the user interfaces can be very different on different computers.

The basic concepts and tools supported by our kernel are mainly intended for researchers and modeling and simulation specialists. Other user categories can be supported by building new user interfaces and new layers of tools. Such tools can allow an architect or a chemical engineer to describe his building or chemical plant and the assumptions in his own language. These kind of tools and user interfaces can be developed as parts of applications projects. Today's "high-level" tools of this kind are too rigid. They produce canned, black box models which cannot be modified. The user is in trouble if some component is missing, since it is very difficult or even impossible for him to add new components. The high-level tools should generate the desired model in an explicit form so that it is readable and can be modified by the user.

3. Kernel specification

This section gives a specification of the kernel to be implemented in the project proposed. For reasons of comprehensibility, it is written in the style of a reference manual as if the kernel already existed.

Introduction

The kernel supports basic data structures for model representation and contain a basic set of tools for model development and simulation.

Priority is given to efficiency and safety. The user and the kernel are considered to work in two modes. The model is built and modified in the model development mode. When the user wants to simulate the model is compiled, which means that the model consistency is checked and an internal, efficient representation for simulation is generated automatically.

For numerical calculations PRO-MATLAB (Moler et al, 1987) is used. It has been extended with routines for numerical solution of differential-algebraic equation systems. The internal representation used at simulation is an M-file. It means that

the representation is readable. There is a tool which translate internal variable references to the user's variable names. This approach does not mean that there is a hard commitment to use PRO-MATLAB. It is fairly simple to modify the back-end of the compilation tool to generate a representation that is adapted to another numerical tool. It is also possible to take the PRO-MATLAB code output by the compilation tool and manipulate that representation.

Below data structures, consistency checks, compilation and simulation tools are in turn discussed in a bit more detail.

Data structures

Besides common data types such as real numbers, vectors and matrices of real numbers the kernel supports a number of model type objects. The basic model type objects are

- model
- terminal
- realization
- parameter
- variable

Model objects have attributes. Some attributes like name, creation time, classification etc are common to all model types. Models have attributes such as terminal list, parameter list and realization list. Simple terminals have attributes like unit of measure, name of quantity, value, value range and structures terminals have a component list. Realizations have attributes such as variable list, submodel list, connections and primitive text (equations).

The model type concept is object oriented. It means that the kernel should support subclass of and super class relations. Two principles of inheritance are needed: union and override. Override inheritance is used for single value attributes. Union inheritance is used for attributes which can have multiple values (list attributes). It means that an object overrides an inherited object with the same name attribute. This inheritance concept allows simple exchange of a specific submodel of a realization as well as change or addition of a terminal component.

The kernel have the tools to create, copy, delete and modify objects. Modifications involve adding and removing components and changing of attribute values. It is possible to store models as text files for later retrieval.

Consistency checks

The kernel contains tools for consistency checks. The basic policy is to allow inconsistent models during the model development phase, since sometimes it is necessary to make a sequence of modifications to change a consistent model to another consistent model.

An extensive consistency check is done in the compilation step, since many of the consistency checks imply a global analysis and deduction of unspecified values from the context. The tools to create and change model objects do not perform extensive consistency checks. They only check simple and local conditions. For example, when connecting two terminals it is easy to check that their explicitly specified attributes are consistent, but it is a bit more laborious to check consistency of unspecified parts since it involves inference from the context. Note that the kernel guarantees that the value of an attribute is of proper type. If this basic requirement is not fulfilled a check has to be performed at each access of attributes. It means that the kernel becomes inefficient.

Many of the procedures for consistency checks could be used in different contexts and therefore are modularized properly to allow reuse. The routines for consistency checks include lexical, syntactical and semantical analysis of equations and consistency checks of connections. There is a database to handle compatibility of quantity names. The consistency and conversion of units of measure is handles by another database.

Compilation

The compilation has two basic purposes. First, it performs extensive consistency checks and second, it transforms the user model to an internal representation adapted to a special task. In this case a the compilation tool transform the model to a representation adapted to simulation.

The compilation is a critical check point and an internal representation for simulation is not generated unless a number of consistency checks are passed.

Simulation tools

The kernel supports simulation of models described by differential-algebraic equation systems. Besides scalar variables also vector and matrix variables are allowed. The kernel does not allow dynamic switches of behaviour descriptions (realizations) during simulation, but realization has to be selected before compilation. The kernel will probably support simulation of models described by difference equations.

The framework is extendable to include polynomials, transfer functions and polynomial matrices. The kernel allows a behaviour description to use functions specified in PRO-MATLAB. This feature allows a user to escape when he has to define complex functions. Note that PRO-MATLAB allows user-specified functions in Fortran and C. Consequently, in this way the kernel allows user-specified functions in Fortran and C.

The tools for setting up a simulation include routines to define parameter values and initial values.

4. Project proposal

We propose a project to implement a kernel for model development and simulation as outlined above. It is a natural step to make the results of the CACE project generally available. The next natural steps are joint application projects with developers and users of CACE tools. However, these application projects assume that there are useful tools available. The purpose of this project is to lay the basis to such tools. The application projects can then allow us to tune the tools and to create new special tools and customized user interfaces for different applications. The kernel can serve as a basis for someone else to develop commercial products.

As the basic implementation language we will use C++ and when feasible we will in different ways use and incorporate available, existing software.

User interface

The kernel has no user interface facilities. However, to make the kernel useful, we will implement some in the project proposed.

In this project we will use the object-oriented interface package Interviews (Linton and Calder, 1987). It is written in C++ on top of the X Window System. InterViews is a public software developed at Stanford.

The textual form for storing models on files is actually one form of a modeling language. Model objects should have capabilities of printing themselves. Printed presentations should have mouse sensible names to access the model object menu.

Different editors should be available:

- A graphical editor for editing the picture (and structure) of a model object.
- A text editor for editing textual representations like equations. Hopefully an external text editor like Emacs can be used.
- Forms for editing objects like terminals with an enumerable number of attributes.

What actually will be implemented depends on what can be done easily in InterViews and what is actually needed to test the kernel and to make the kernel useful. We will also investigate the possibilities to use user interface facilities of PRO-MATLAB. If possible we will make an interface to SattGraph 1000 to allow animated presentations of the simulation results.

Guest researcher

We have good contacts with the groups at AT&T Bell Laboratories responsible for the development of C++. Andrew Koenig is a Distinguished Member of the Technical Staff at AT&T Bell Labs. He is working on C++ tools and has participated in the development of version 2 of C++. He has visited us and is interested in our project. Andrew is also writing a book on C++ and is interested in giving a series of lectures. A person with his experiences of using C++ would be very useful in the beginning of our project. He would share his wealth of experiences developing in C++ and could help us to design the basic architecture of the kernel correctly. In preliminary discussions he has expressed interest to visit us 1 - 2 months this summer.

Proposal budget

Funds requested from STU. The amounts are in 1000 SEK.

	Total	1989/90	1990/91
Personnel	1 145	765	380
Materials	90	60	30
Travel	60	40	20
Computer and software service	85	55	30
Administration costs (8%)	120	80	40
Amount of this request	1 500	1 000	500

The personnel costs correspond to 4 man years.

5. References

ANDERSSON, M. (1989): "An Object-Oriented Modeling Environment," *Proceedings of the 1989 European Simulation Multiconference*, Rome, June 7-9, 1989.

- LINTON, M. and P.R. CALDER (1987): "The Design and Implementation of InterViews," *Proc. USENIX C++ Workshop*, November 9-10, 1987, Santa Fe, NM, USA.
- MATTSSON, S.E. (1988): "On Model Structuring Concepts," *Preprints of the 4th IFAC Symposium on Computer-Aided Design in Control Systems (CADCS)*, August 23-25 1988, P.R. China, pp. 269-274.
- MATTSSON, S.E. (1989): "Modeling of Interactions between Submodels," *Proceedings of the 1989 European Simulation Multiconference*, Rome, June 7-9, 1989.
- MOLER, C., J. LITTLE and S. BANGERT (1987): *PRO-MATLAB: User's Guide*, The MathWorks, Inc., Sherborn, MA, USA.
- STRAUSS, J.C. (Ed.) (1967): "The SCi Continuous System Simulation Language (CSSL)," *Simulation*, Dec 1967, 281-303.
- WOLFRAM, S. (1988): *MathematicaTM: A System for Doing Mathematics*, Addison-Wesley.

Anvisningar - Se broschyren »Sök stöd hos STU«. Om sekretessbeläggning önskas av vissa uppgifter bör dessa föras samman i en särskild bilaga som förses med påskrift »Sekretessbeläggning önskas«. Läs vidare om detta i broschyren.

ANSÖKAN OM PROJEKTANSLAG

Datum
1989-03-30

Ansökan avser	tidigare diariet/projekt nr	Sekretessbeläggning önskas av bilaga nr
<input checked="" type="checkbox"/> Nytt projekt	<input type="checkbox"/> Fortsättning på tidigare STU-projekt →	

Sökande	Namn (institution/företag/enskilda personer) postadress och postgironummer)		Tel.nr under arb.tid (även riktnr)	Antal anställda	Länsbokstav		
	Sven Erik Mattsson Institutionen för Reglerteknik Lunds Tekniska Högskola Box 118 221 00 LUND pg 1 56 50 - 5		046-108779	30	M		
	Organisationsnummer/Personnummer						
Koncern tillhörighet (koncernens namn, gäller endast då sökanden är företag)		antal anställda i koncernen <input type="checkbox"/> under 500 <input type="checkbox"/> över 500					
Projektledare	Namn och telefonnr (även riktnr) Sven Erik Mattsson 046-108779						
Projekttitel	Kärna för modellering och simulering						
Tid- och kostnadsplan	Arbetet utföres vid Inst för Reglerteknik, LTH		och pågår under tiden (fr o m - t o m) 890701 - 910630				
Kostnadsslag	Projektets totala kostnad	Totalt	Ansökan om projektanslag från STU				
			Totalkostnad fördelad per budgetår (07-01 - 06-30)				
			198 9 / 90	198 90 / 91	198 /	198 /	198 /
Löner/Eget arbete Procent	769 000	769 000	514 000	255 000			
Lönebikostnader ¹ 49	376 000	376 000	251 000	125 000			
Konsultkostnader/Legoarbeten							
Utrustning							
Material	90 000	90 000	60 000	30 000			
Resor inkl traktamenten	60 000	60 000	40 000	20 000			
Patentkostnader Procent							
Förvaltningsavgift ² 8	120 000	120 000	80 000	40 000			
Levnadskostnader ³							
Datorunderhåll	85 000	85 000	55 000	30 000			
Övrigt - SPECIFICERA!							
Summa	1 500 000	1 500 000	1 000 000	500 000			

¹ Socialförsäkringsavgifter m m (för statliga institutioner f n 39 % av lönesumman)

² Gäller endast universitet, högskolor ³ Kostnader för eget arbete



Annan finansiering	<p style="text-align: right;">Myndighet, anslagsgivande organ e d</p> <p>1 Beloppet som söks hos STU söks även hos Belopp</p> <p>2 Utöver det belopp som söks hos STU behövs för detta projekt ytterligare kronor, vilket söks hos</p> <p>finansieras av</p>
Patentfrågor m m	<p>Patent och patentansökningar nr</p> <p>Ägare till projektet Namn</p> <p><input type="checkbox"/> Sökanden <input type="checkbox"/> Annan anges</p>
Internationellt samarbete	<p>Ingår internationellt samarbete? Ange land/samarbetspart.....</p> <p>Bifoga gärna ett kort referat på engelska</p>
Projektbeskrivning <small>(Mål, metod/teknisk lösning, dagläge och resultat-användning)</small>	<p>Intresset från industrin för modellutveckling och simulering är stort. Man vill kunna designa och köra processerna effektivare och skapa en bättre miljö för processoperatörerna. Dagens verktyg för modellutveckling och simulering uppfyller inte kraven. I CACE-projektet har det utarbetats förslag till hur en ny generation verktyg bör se ut. Erfarenheterna från prototypen indikerar att idéerna är sunda och att den kan användas som bas för en ny generation simulatorer. Prototypen, som är skriven i Common Lisp och KEE, förväntas inte få någon större spridning, då KEE är dyrt. KEE har använts då det möjliggjort implementering med moderata insatser.</p> <p>Vi föreslår ett projekt för att implementera en kärna för modellutveckling och simulering enligt den specifikation som framtagits i CACE-projektet. Detta är ett naturligt första steg i att föra ut resultaten från CACE-projektet, som sedan kan följas av tillämpningsprojekt. Med en kärna menas rutiner för att handha och manipulera den interna representationen. I vår design finns en klar separation mellan användargränssnittet och kärnan.</p> <p>Målet är att implementera en kärna som av någon annan kan utvecklas vidare till en kommersiell produkt. Vi vill fokusera insatsen på en kärna och bara göra ett enkelt användargränssnitt för att hålla projektet på för oss en rimlig storlek. Det är arbetskrävande att implementera avancerade användargränssnitt. Kärnan skall vara användbar för att visa och illustrera våra idéer, vilket vi kan göra med enklare användargränssnitt. Den skall också vara användbar i tillämpningsprojekt med användare av dessa verktyg. Där kan det vara naturligt att tillverka anpassade användargränssnitt.</p> <p>C++ kommer att vara det grundläggande implementeringspråket. För de numeriska beräkningarna avser vi använda PRO-MATLAB. För övrigt kommer vi att använda färdiga programkomponenter där det är lämpligt.</p> <p>Nyckelord:.....</p> <p style="text-align: center;">modellering, simulering, programvara</p>
Förhandskontakter på STU	<p>Har denna ansökan diskuterats med någon handläggare på STU?</p>
Underskrifter	<p>Sökande och projektledare (namnteckning med namnförtydligande)</p> <p style="text-align: center;"><i>Sven Erik Mattsson</i></p> <p>Sven Erik Mattsson</p>
	<p>Firmatecknare eller prefekt/föreståndare (namnteckning med namnförtydligande)</p> <p style="text-align: center;"><i>Björn Wittenmark</i></p> <p>Björn Wittenmark</p>

P R O T O K O L L

från telefonmöte med STUs styrgrupp för ramprogram CACE den 13 juni 1989 kl 08 30 - 9 30.

Deltagare: Sven Gunnar Edlund
 Karl Eklund
 Sven Erik Mattsson
 Gustaf Söderlind
 Arne Otteblad

- § 1 Formalia
 Ordförande: Sven Gunnar Edlund
 Sekreterare: Arne Otteblad
 Justeringsman: Sven Erik Mattsson
- § 2 Den huvudsakliga punkten på dagordningen var programmet för höstens CACE-seminarium.

Efter intensiva diskussioner enades man om ett förslag där poster-sessionen låg på båda sidor om lunch-pausen och paneldiskussionen startade kl 15 med avslutning 16 30.

Sven Gunnar E åtog sig att vara ordförande för seminariet medan Karl E åtog sig att leda paneldiskussionen.

Avgiften för seminariet bestämdes till 450 kr.

Förslaget till utformningen av inbjudningen till seminariet och presentationstexten godkändes varefter Sven Erik M och Arne O fick i uppdrag att arbeta vidare med de praktiska arrangemangen.

- § 3 Nästa styrgruppsmöte bestämdes till den 8 september i Lund. Då skall projektverksamheten presenteras.

Vid protokollet



Arne Otteblad
 STU

Justeras



Sven Erik Mattsson
 LTH

Dokumentation i form av rapporter och kopior av overheadbilder kommer att delas ut till deltagarna.

Avgiften är 450 kr, vilket inkluderar lunch, kaffe och dokumentation.

Avgiften sätts in på **postgiro**nummer 18 31 11-4. Var god ange att inbetalningen skall **bokföras redovisningsnr. 89-2110 PK**.

Anmälan och avgiften bör vara **STU tillhanda senast den 6 oktober 1989** på bifogade svarskort eller per telefon 08-775 44 53, Kerstin Bohs.

Antalet deltagare är begränsat. I händelse av för många anmälningar får de först anmälda förtur. Vi bekräftar Ert deltagande senast den 18 oktober 1989.

Närmare information lämnas av Sven Erik Mattsson, LTH, (046-10 87 79), eller Arne Otteblad, STU, (08-775 44 52)



THE SWEDISH NATIONAL BOARD FOR TECHNICAL DEVELOPMENT
BOX 43200 S-10072 STOCKHOLM SWEDEN PHONE +46 8 775 4000
OFFICE LILJEHOLMSVÄGEN 32

Nya verktyg för modellutveckling och simulering

Seminarium
Tisdagen den 24 oktober 1989
Hörsalen, STU
Liljeholmsvägen 32
Stockholm

Syfte: Att sprida kunskap om resultat från ett av STUs ramprogram.
Målgrupp: De som är sysselsatta med eller är ansvariga för utveckling och underhåll av reglersystem och processanalys. Seminariet vänder sig också till dem som är intresserade av datorstöd för ingenjörsarbete i bredare mening liksom de som är intresserade av modellutveckling och simulering i allmänhet.

STU
Styrelsen för teknisk utveckling

Institutionen för reglerteknik
Lunds tekniska högskola

För att kunna utveckla bra tekniska system behövs datorbaserade verktyg för modellutveckling, simulering, analys, konstruktion, uppbyggnad och utprovning. Dagens verktyg har visat sig vara effektiva och arbetsbesparande, men de är mer anpassade till de begränsningar som datorerna hade för 10-20 år sedan än till användarnas önskemål.

För fem år sedan startades STU-ramprogrammet "Datorbaserade hjälpmedel för utveckling av styrsystem (Computer Aided Control Engineering, CACE)" med institutionen för reglerteknik vid Lunds Tekniska Högskola som huvudaktör. Målet var att utforska vilka möjligheter de senaste årens snabba utveckling inom informationsteknologin ger att förbättra de datorbaserade verktygen och att utveckla några prototypsystem.

Den första fasens pilotprojekt visade att datorgrafik, expertsystemteknik och symbolisk manipulering på ett märkbart sätt kan förbättra dagens verktyg. Man kan göra verktyg som tillåter användaren att definiera sitt problem på ett för honom naturligt sätt och förser honom med bättre stöd och hjälp.

Alla metoder för analys och design behöver någon modell av systemet och representation av modeller är därför centralt. Den sista fasen av CACE-projektet har därför fokuserats på ett förslag till en ny generation verktyg för modellutveckling och simulering. Förslaget definierar en kärna för modellrepresentation som bör vara intressant för alla som använder modeller. En grundläggande idé är att stödja återanvändning av modeller så att modeller kan användas för olika uppgifter och lätt anpassas till att beskriva liknande processer, eftersom det är svårt och arbetsamt att utveckla nya modeller. Kärnan tillåter användaranpassade gränssnitt och verktyg så att olika användare kan beskriva sina system och problem i sina termer.

Som avslutning av ramprogrammet inbjuds till en informationsdag där förmiddagen ägnas åt att presentera resultaten. Posters kommer att presentera resultaten från den första fasens pilotprojekt. Eftermiddagen ägnas åt att blicka framåt. För att göra resultaten mer allmänt tillgängliga har ett STU-projekt för att implementera kärnan i C++ startats. Tillämpningsprojekt för att testa och föra ut idéerna planeras.

Anmälan till seminariet: se sista sidan och bifogade svarskort

PROGRAM

Tisdag 24 oktober 1989

Kl 09.00-16.30

Ordförande: Sven Gunnar Edlund

09.00	Registrering och kaffe
09.30	Introduktion Avd chef Bo Stenviken, STU
09.40	Simulering - En aktuell teknik Arne Otteblad, STU
10.00	CACE-projektet - en översikt Sven Erik Mattsson, Inst för reglerteknik, LTH
10.15	SEE - en objektorienterad miljö för modellutveckling Sven Erik Mattsson och Mats Andersson, Institutionen för reglerteknik, LTH
12.00	Posters och demonstrationer Hierarkiska blockdiagram med informationszoomning Intelligent hjälpsystem till dataanalysprogrammet Idpac Symbolisk manipulering av linjära system i Macsyma Ett expertsystem för reglering på lokal nivå Steglängdsreglering vid lösning av differentialekvationer Implementeringsspråk och grafiksprogram
12.30	Lunch
13.30	Posters och demonstrationer (fortsättning)
14.00	Tillämpning: Modellering av kemiska processer Bernt Nilsson, Inst för reglerteknik, LTH
14.30	Differential-algebraiska system Gustaf Söderlind, ITM
14.45	Kaffe
15.00	Paneldiskussion: Framtida behov för modellutveckling och simulering Karl Eklund, KEAB AB (ordf) Sven Gunnar Edlund, STFI Claes Källström, SSPA Sven- Erik Mattsson, Inst för reglerteknik, LTH Staffan Nordmark, Statens Väg- och Trafikinstitut Ann-Britt Östberg, Sydkraft
16.30	Avslutning

Anmälan

Jag anmäler mig till seminariet
Nya verktyg för modellutveckling och simulering
på STU, Stockholm
tisdagen den 24 oktober 1989

Namn (textat):

Arbetsplats:

Adress:

Tel:

**Anmälan insändes till Styrelsen för teknisk utveckling, att: Kerstin Bohs, Box 43200,
100 72 Stockholm**

Senast fredagen den 6 oktober 1989

Porto

Styrelsen för teknisk utveckling
Att: Kerstin Bohs
Box 43 200
100 72 STOCKHOLM

PROTOKOLL

från sammanträde med STUs styrgrupp för ramprogram CACE
den 8 september 1989 kl 10 00 - 16 00

Deltagare: Sven Gunnar Edlund (deltog per tel i §§ 1,3,4,5)
 Karl Eklund
 Sven Erik Mattsson
 Gustaf Söderlind
 Arne Otteblad

§ 1 Formalia

Ordförande: Sven Gunnar Edlund
 Sekreterare: Arne Otteblad
 Justeringsman: Sven Erik Mattsson

§ 2 Sven Erik Mattsson och hans medarbetare presenterade och demonstrerade på sedvanligt utmärkta sätt verksamheten och resultaten inom programmet.

I detta sammanhang finns det då anledning att speciellt notera följande från genomgången:

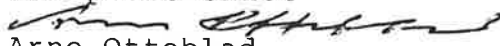
- a) De medel som beviljats inom ramen för CACE-ramprogrammet var förbrukade den 31 augusti. Den fortsatta verksamheten med implementeringsprojektet finansieras nu med andra medel från STU i enlighet med styrgruppens tidigare rekommendation.
- b) Som resultat av ramprogrammet har hittills följande resultat framkommit: 1 doktorsavhandling, 3 lic avhandlingar, 8 examensarbeten, 5 vetenskapliga artiklar, 23 konferensbidrag, 8 rapporter och 2 STU-seminarier. Sett i proportion till den totala kostnadsramen för programmet klarar resultaten väl en jämförelse med andra ramprogram på STU.

§ 3 Inför seminariet den 24 oktober diskuterades speciellt några detaljer rörande dokumentationen och organisationen av paneldiskussionen.

§ 4 Protokollen från telefonmötena den 24 februari, 6 april och 13 juni 1989 godkändes och lades till handlingarna.

§ 5 I och med seminariet den 24 oktober är styrgruppens formella uppdrag avslutat. Sven Erik Mattsson och Arne Otteblad ansåg det som mycket värdefullt om gruppen kunde fortsätta som rådgivare i det fortsatta implementeringsprojektet. Då samtliga ansåg det vara intressant enades man om att fortsätta som en projektreferensgrupp och bestämde att ha nästa möte i Lund den 16 maj 1990.

Vid protokollet


 Arne Otteblad
 STU

Justeras

 Sven Erik Mattsson
 LTH