Research Issues in the Areas of Programming Languages and Communication Protocols: My Personal Experience since the 1970ies

Gregor v. Bochmann

School of Electrical Engineering and Computer Science

University of Ottawa (Canada)

Presentation originally given (in french) at IRISA, Rennes (France)

March 14, 2012

Abstract

- **After my PhD in theoretical physics in 1971, I started to work in computer science. The choice of a particular research area was not so easy. In the area of programming languages, much work was done at that time on compiler writing systems. Since the syntactic aspects of programming languages were already well explored, I worked during several years on the formalization of the semantic aspects of languages by adapting the concept of semantic attributes to compilation in several passes. Then I was looking for a research area that was less developed at the time, and in 1974, I started work on communication protocols. This was the time of the first experiments with computer networks and the concept of protocols was still quite vague. The work I did, together with colleagues from different countries, on the specification, verification, implementation and test of protocols laid the foundation for the systematic development of communication protocols and their standardization.** In this talk, I will also discuss the impact of these research results on today's state of the art in this area.

Overview

- Historical perspective – the factors of development
- My personal itinerary
- Research issues
  - for programming languages
  - for communication protocols
  - for model-based system development
- Recurring research themes
- Comments - teaching, collaborations
- A kind of conclusion

Historical perspective of computer science

- **The 1950ies**
  - First high-level programming languages: Fortran, Cobol, Algol, Lisp
- **The 1960ies**
  - Theories of languages, compilers
  - Operating systems, shared resources, time sharing
- **The 1970ies**
  - Computer networks, remote access to computers
  - Languages for programming concurrent processes
- **The 1980ies**
  - Object-oriented design and programming
  - Multimedia system – hypermedia (Videotex, telephone directories, etc.)
  - Artificial intelligence (5th generation computers in Japan, expert systems, etc.)
  - OSI (Open System Interconnection) protocol standardization

Historical perspective of computer science (2)

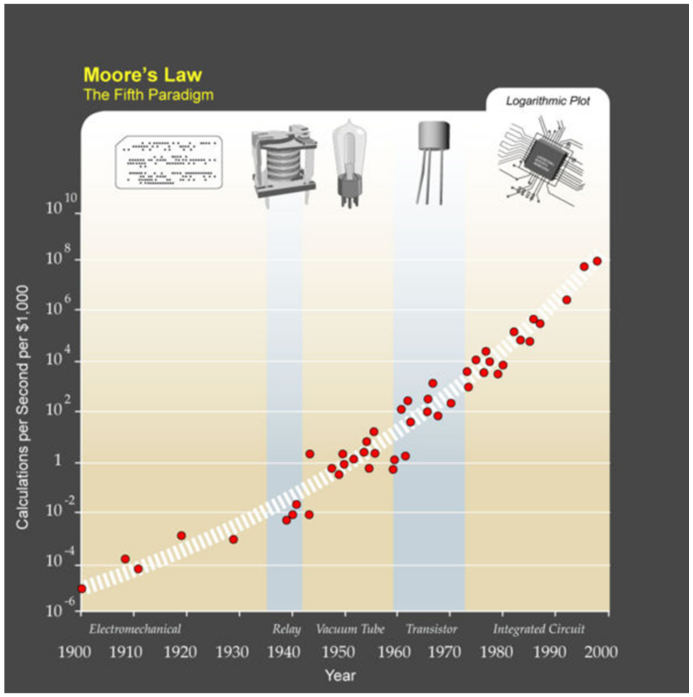
- **The 1990ies**
  - Large-scale utilisation of the Internet
    - WWW
    - E-mail service
  - UML
- **The 2000ies**
  - Web-2 applications – Web Services
  - Mobility – wireless networks
  - Peer-to-peer systems – cloud computing
  - Open Source development

Factors of development

Technological advances

- Moore’s law concerning the speed of processors

Calculations per second for 1000$



Osborne:
the first portable computer
1982

- Similar development for the transmission bandwidth of digital networks

Factors of development (2)

Abstraction – interfaces

- Procedure call interfaces
  - e.g. object orientation
  - also remote procedure calls (RPC)
- Component-based design, re-use
- Layered system architectures
  - operating systems
  - communication protocols

Factors of development (3)

Virtualisation – standardization

Standardized functional interfaces, variable performance, flexible use in different contexts.

Examples:

- Virtual machine, e.g. P-code for Pascal [Wirth 1972], Byte-code for Java [1995] – can be interpreted by different computers
- Virtual memory (degraded performance compared to real memory)
- Virtual connection – communication service (e.g. virtual call of X.25 [1976])
- Virtual terminal (e.g. X.25 PAD, Teletex terminal, web browser)
- Computer emulation (degraded performance)
- Cloud computing

Factors of development (4)

Virtualisation – standardization(2)

Standardization bodies:

- International (e.g. ISO, ITU-T, IET??) and national organizations
- US organizations with international participation : e.g. IEEE, Internet Society
- Industry groups, e.g. OMG
- ad hoc standardization by market domination: e.g. the PC of IBM - the Internet protocols - Windows of Microsoft - and in the future ??

Impact of standardization:

- facilitates re-use
- leads to compatibility
- increases the competition among manufacturers
  - therefore reduction of costs and multiplications of applications

My personal itinerary

Different residences (in Germany)

- During and after the war, my mother was looking for a permanent home in Schleswig-Holstein: Kiel – 1943 – Selent – 1946 – Möltenort – 1949 – Krummsee – 1955 – Eutin-Fissau – 1970

Differents universities: studies in physics

- **Kiel** (1961-62) ; **Tübingen** (1962-63) ; **Lübeck** (one year of music studies - piano and cello, 1963-64); **Grenoble** (1964-65); I lost one year through a hepatitis (1965-66); **Munich** (1966-67) ; Master project at CERN (**Genève**, 1967-68); doctoral studies at McGill University in **Montréal** (1969-71)

Differents areas of research

- Master project: Experimental physics of elementary particles (programming a PDP-8 computer for real-time data analysis)
- PhD project: simulation studies of reactions of elementary particles at high energy on nuclei (theoretical physics requiring much programming in languages such as Fortran, PL-1 and Formac)
- Post-doctoral research project in computer science at the University of Montréal (French speaking university – 1971-72)
- Professor in computer science at the University of Montréal (1972- 1997), and University of Ottawa (bi-lingual university – since 1998)

My personal itinerary (2)

Research areas in computer science

- Reasons for choosing computer science: I was looking for a research area with less professional competition than in high-energy particle physics, and with more applications useful to mankind.

Neural networks (1971-72) with Bill Armstrong

- I found heuristics, but no fundamental laws – and the practical results were mixed

Compiler writing systems (1972 until around 1976) with Olivier Lecarme

- I liked the syntactic rules clearly specified, and the corresponding analysis algorithms
- I developed algorithms for analyzing the semantics of languages to be processed by a multi-pass compiler
- I regretted that the basic principles in this area were already well explored

Computer communications and networks (from around 1974)

- This is the area in which I eared my largest recognition (see below)
- During the 1980ies, there was much industrial interest in this area because of the new technological developments and the OSI standardization efforts
- From early on, I found financial support for my work from the Department of Communications of the Government of Canada, and later from the industrial partners of the Centre de Recherche Informatique de Montréal (CRIM) and from Hewlett-Packard, the industrial partner of my Industrial Research Chair at the University of Montreal

My personal itinerary (3)

Other research areas (since around 1995)

- Quality of service negotiation (1995 – 2006)
  - for access to multimedia databases and load balancing among servers
  - for conversational applications (e.g. téléconférences) and mobility
- Control protocols for all-photonic networks and their performance
  - I was the coordinator of the research axis “network architectures” in the Research Network “Agile All-Photonic Networks” including 15 professors from 5 universities and 5 companies (2003 – 2008)
- Peer-to-peer systems (since 2005)
  - Load balancing
  - Real-time video streaming
  - Search in data bases with weak consistency
- Crawling and modeling Rich Internet Applications (Web-2) (ongoing)
  - Project in collaboration with IBM

Research issues in the area of programming languages

Concepts of programming languages

- Procedure calls and recursion (Algol 60, LISP)
- Object-orientation and garbage collection
  Simula 67, Smalltalk 80, “hype” in the 90ies, now e.g. Java
- Structured programming (Dijkstra’s paper “goto considered harmful”)
  - My short article “Multiple exits from a loop without the goto” (1973)
- Very high level languages (APL, Setl)
  - I got interested in modeling and specification languages

Research issues in the area of programming languages (2)

Description techniques and language processing

- Grammars
- Lexical and syntax analysis
- Compiler writing systems (automation)
  - my first Master project supervised: a CWS for LL(1) grammars – and still today, I teach this subject in an undergraduate course
- Definition of the semantics of languages

Description techniques and language processing (2)

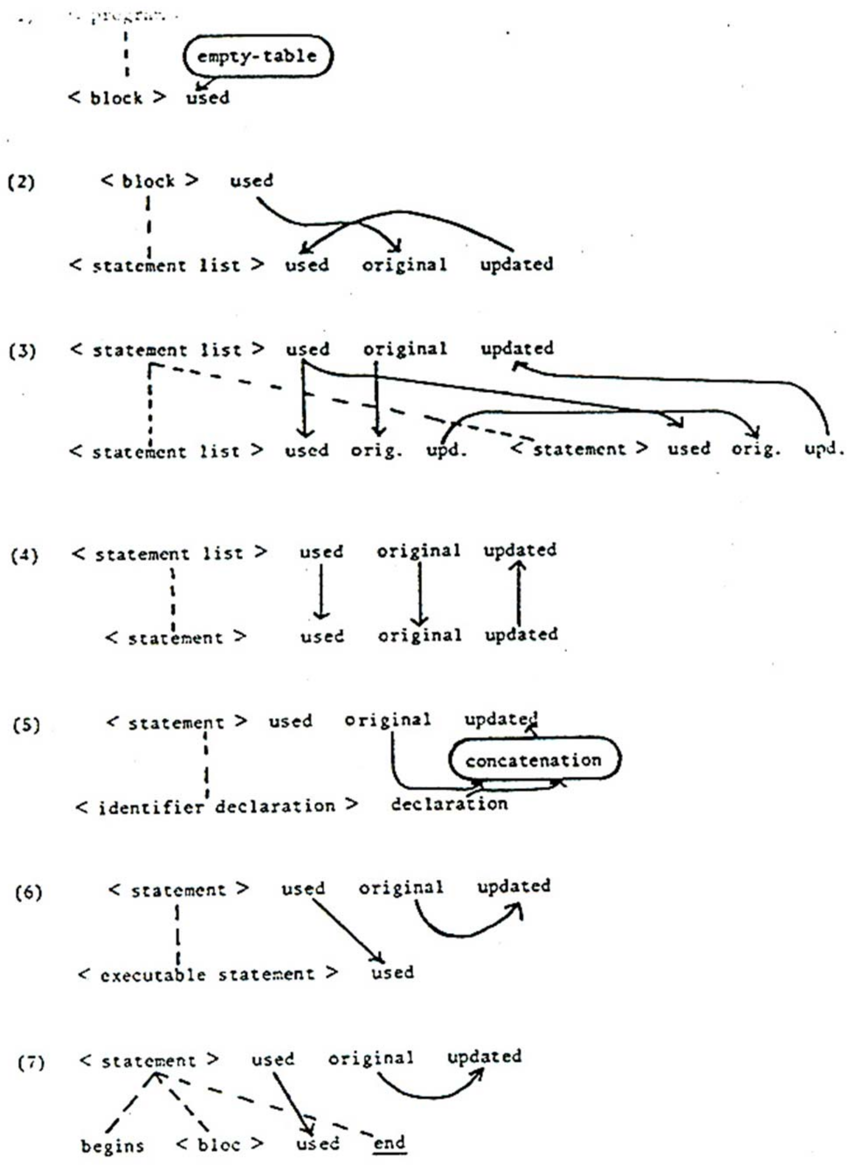
Semantic language definition using Semantic Attributes (proposed by Knuth in 1968)

- Open Question: In which order can the semantic attributes on the syntax tree of a given program be evaluated ?
- Often, multi-pass compilation was used
- In my paper of 1976, I explain under what conditions the attributes can be evaluated in a fixed number of passes through the program from left to right
  - These conditions have been used by many compiler writing systems that have been developed subsequently by other people
  - I found these conditions again in a recent text book (by Sebesta) which I used in a course on the concepts of programming languages

Evaluation of semantic attributes : an example

- Evaluation rules

  One evaluation rule per syntactic production rule (copied from my paper of 1976)

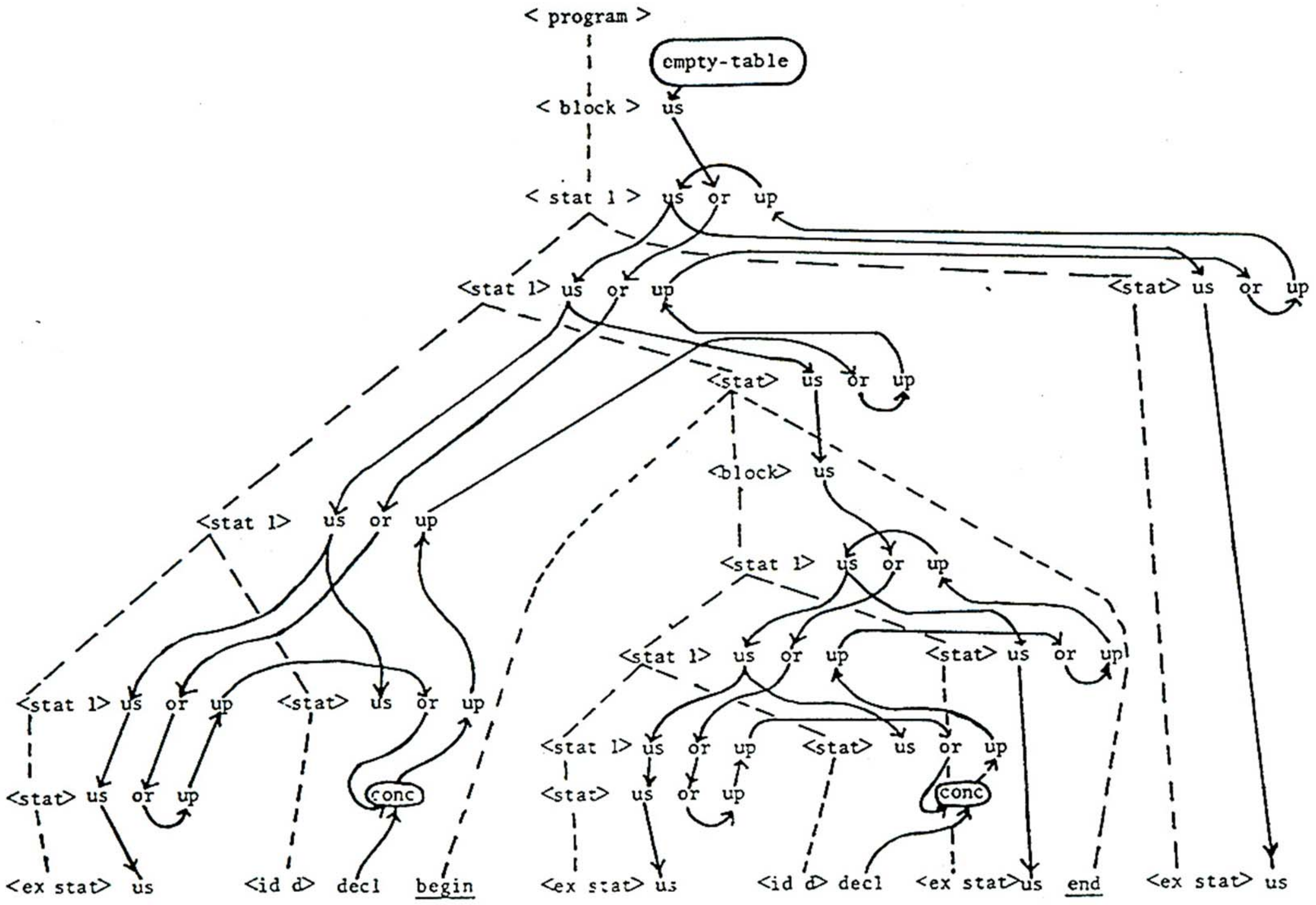


program
empty-table
< block > used
(2) < block > used
< statement list > used original updated
(3) < statement list > used original updated
< statement list > used orig. upd. < statement > used orig. upd.
(4) < statement list > used original updated
< statement > used original updated
(5) < statement > used original updated
concatenation
< identifier declaration > declaration
(6) < statement > used original updated
< executable statement > used
(7) < statement > used original updated
begins < bloc > used end

Evaluation of semantic attributes : an example

- The syntax tree of a program

  (with functional dependencies of attributes)

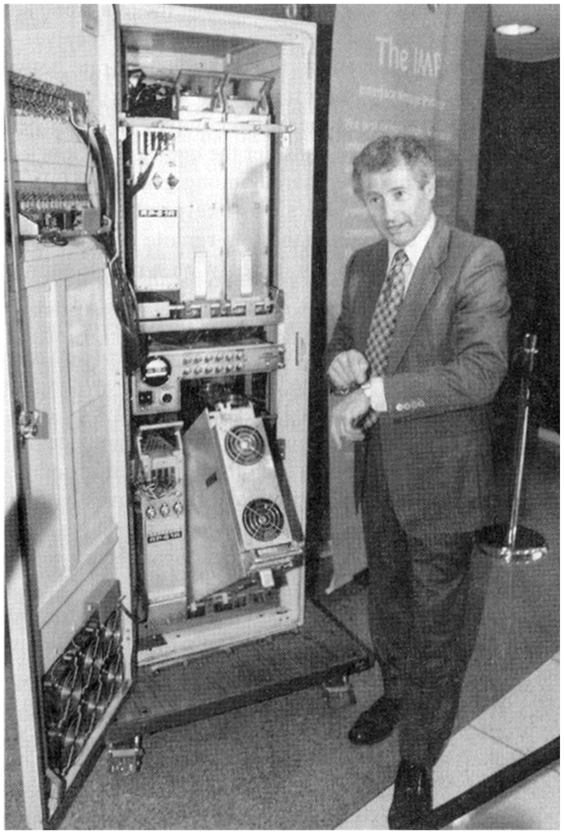


Research issues in the area of communication protocols

Computer networks – during the 1970ies

The first experimental networks

ARPANET (USA): long distance network – 1969
NPL network (UK): first LAN
Cyclade (France): long distance network – 1972
- introduction of a datagram protocol, precursor of the IP



Leonard Kleinrock, UCLA with an ARPAnet node

Donald Davies, NPL

Louis Pouzin INRIA (France)

Research issues in the area of communication protocols (2)

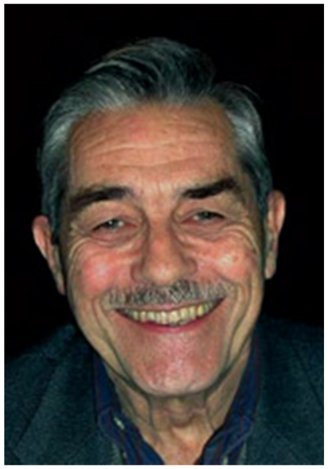
Development of protocoles – the years ‘70 – ‘80

- Protocol standards
  - X.25 : first protocol standard for computer networks
- Protocol architectures of manufacturers
  - IBM (SNA), DEC, Honeywell, etc.
- Application protocols
  - Internet: e.g. FTP et SMTP (in the 1970ies)
  - Videotex – first version of a WWW (in France: terminal for telephone directory – around 1980)
  - ASN.1 and OSI “Remote Operations” (around 1984) : similar to the “Web Services” of today

My personal history

I met Louis Pouzin at a conference in 1973

I analyzed the alternating bit protocol (ABP) in 1974 and developed the reachability analysis method for verifying protocols modeled as state machines – article in 1975



I applied the same method for the verification of the X.25 protocol (journal paper in 1978)

I also experimented with program proof techniques to verify protocols (article in 1975)

My personal history (2)

- In 1977, with Jan Gecsei, I proposed the modeling with extended state machines

  When I presented the paper at the IFIP Congres in Toronto, I met Zafiropulo from the IBM research lab in Zurich who worked with Colin West and Harry Rudin on the verification of protocols.

- I worked with Carl Sunshine on the formalization of:
  - **Protocol**: not defined as an interface over distance between two entities, but defined as the behavioral requirements to be satisfied by each entity
  - **Service**: an abstraction of a distributed system layer which contains several protocol entities

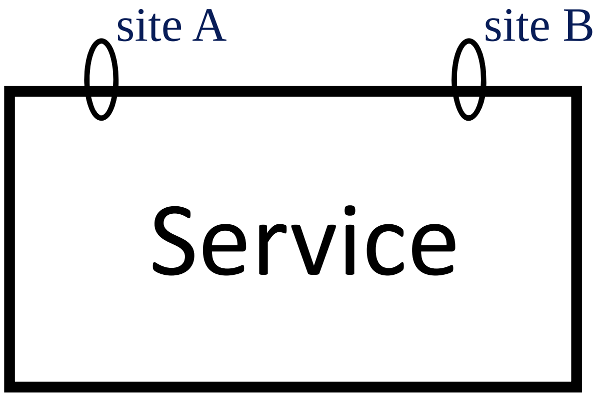
What is a communication service – – what is a communication protocol ?

How to specify a service ?

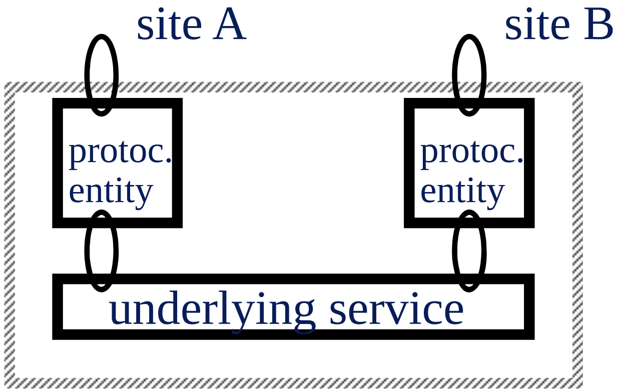
- *It is a system component distributed over several different sites*
- **Local properties** for each interface, and in addition **global properties**
- **Abstract interactions** at the interfaces in **both directions** (in contrast to the interface of an « objet »
- NOTE: This concept was generalized with Michel Raynal for arbitrary components

How to specify a protocol ?

- An **abstract model** of the behavior of an entity with **two interfaces**
- Message exchanges – definition of the **encodings in detail**



Communication Service



Communication Protocol : two protocol entities

Research issues in the area of model-based system development

- **This area has been developed by the researchers working on communication protocols already in the 1980ies and before.**
- **In the 1990ies, this area has become very popular with the development of the UML notation** *– more than 10 years later*

Research issues in the area of model-based system development (2)

A comparison:

Protocol development	Model-based development
• **Protocol specification**	• **Model of the system**
• Protocol verification • Protocol implementation • Conformance testing of an implementation (tests based on the specification – ”black-box testing”)	• model checking • Model transformation • “model-based testing”

Research issues in the area of model-based system development (3)

A comparison (2)

Specification languages - tools (automation) for

- **editing the specification,**
- **verification,**
- **implementation,**
- **test.**

Protocol development	**Model-based development**
• Formal Description Techniques (FDTs) for protocols and services OSI (during the 1980ies) – LOTOS (tools from unversities) – Estelle (e.g. tool from INT, France) – SDL (tools: e.g. Geode from Vérilog, Tau from Telelogic-IBM)	• UML v1 (1996) • UML v2 – SDL (2005)

Research issues in the area of model-based system development (4)

Some results from our research

- Reachability analysis
  - For a distributed system, the step before “model checking”
- Implementation code generation from model specifications (modeling languages Estelle, ASN.1)
- Generation of test suites from model specifications
  - from 1989 to 1997 in the context of an industrial research chair at the University of Montréal
- Mondel: an object-oriented specification language

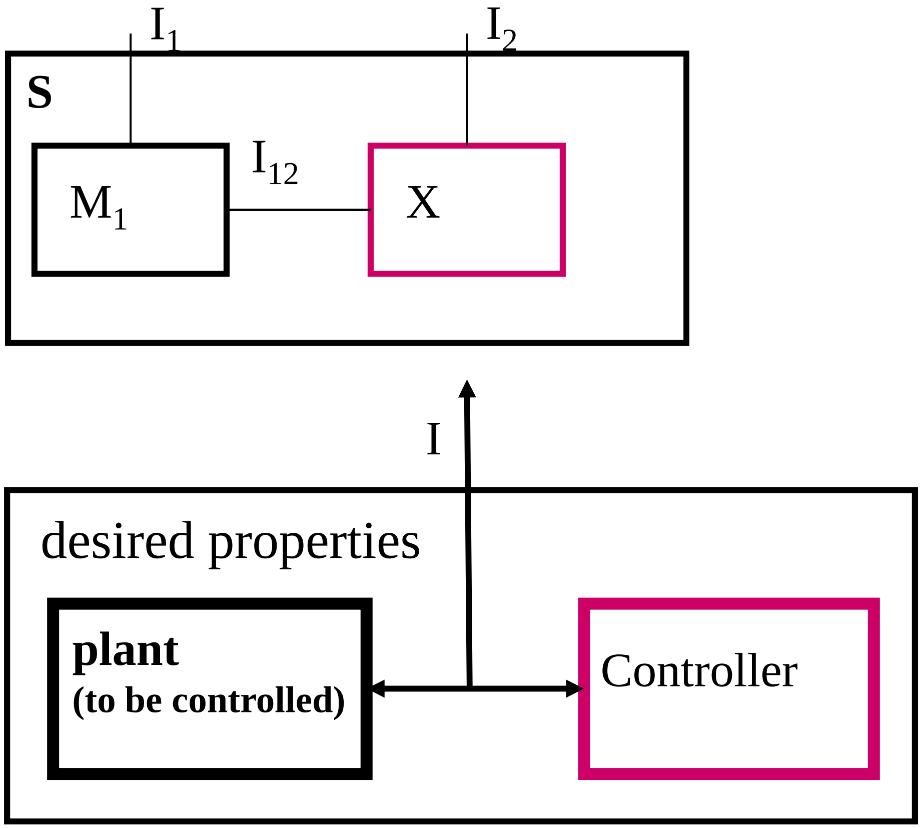
Recurring research themes:
Behavior derivation for sub-modules

The problem:

Find the specification for module X, given the behaviors of S and M_1

The applications:

- Design of a controller

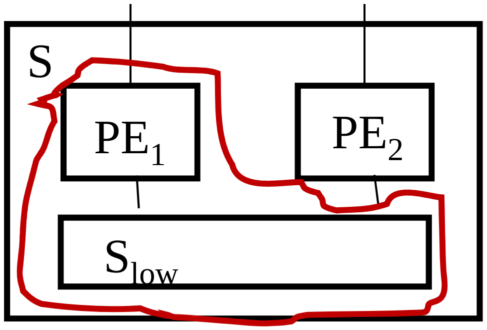


Recurring research themes :

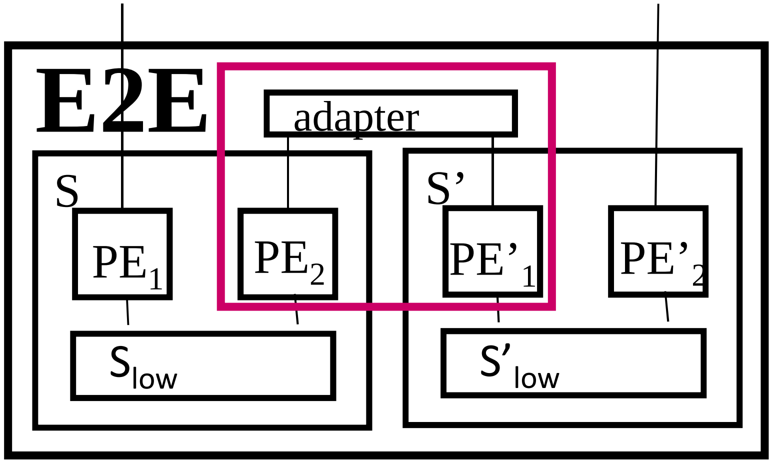
Behavior derivation for sub-modules (2)

The applications (suite):

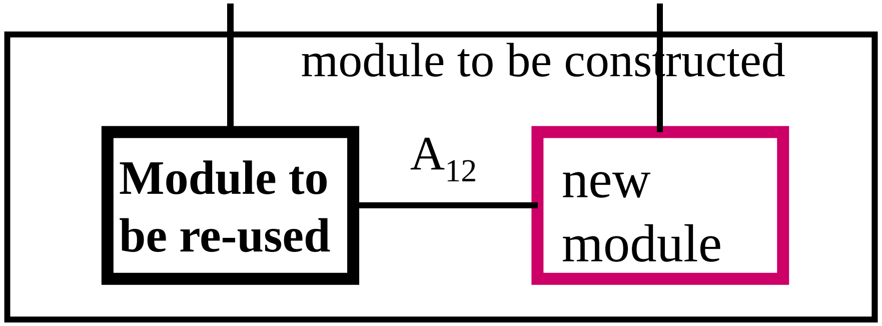
- Protocol design



- Design of a protocol converter



- Re-use of components

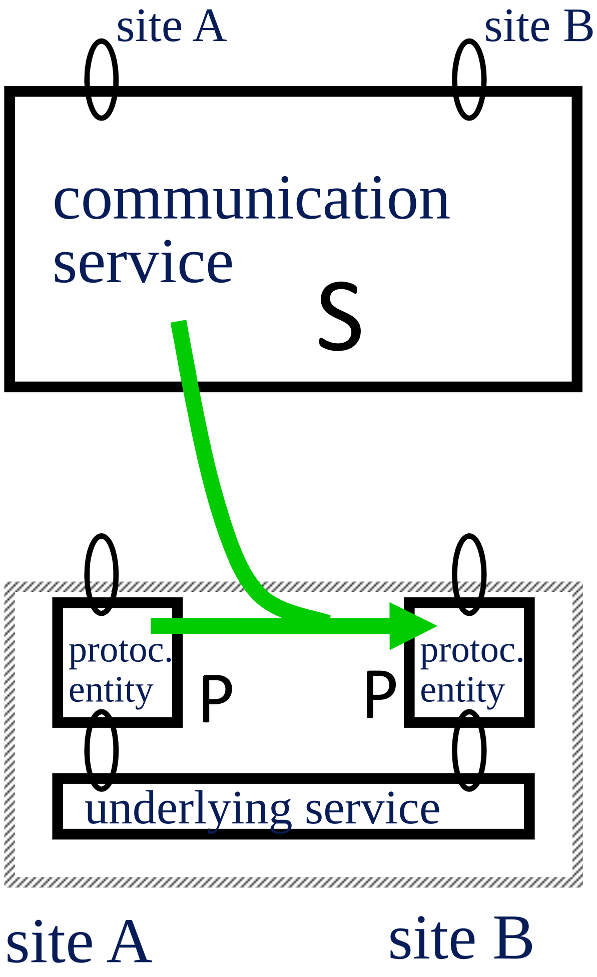


Recurring research themes :

Behavior derivation for sub-modules (3)

- The initial idea with Philip Merlin in 1980
  - behavior specification in the form of asynchronous automata
  - for input-output automata and other conformance relations (in the 1990ies with PhD students Drissi and Tao)
- An observation by Nina Yevtushenko in 2000:
  - similarity of the formulas for synchronous and asychronous automata
- Generalization of the problem and solution
  - formulation of the problem in the context of data bases (2001)
  - formulation of the problem in the context of first-order logic and adaptation of the solution to the context of synchronous and asynchronous automata, communicating by rendezvous or by input-output interactions (2011)

Recurring research themes:
Protocol derivation from a given service specification



Historical notes:

- 1978: the meaning of

“a protocole P offers the service S”

(Finite State Description of Communication Protocols)

- 1980: behavior derivation for sub-modules (with Merlin)

Recurring research themes: **Protocol derivation from a given service specification**

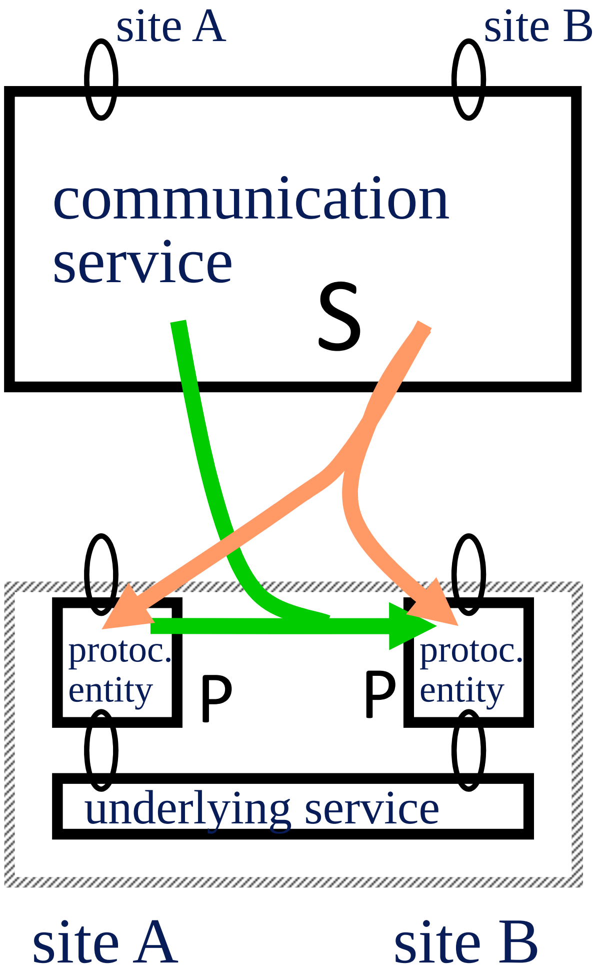
Historical notes:

- 1978: the meaning of

  “a protocole P offers the service S”

  (Finite State Description of Communication Protocols)

- 1980: behavior derivation for sub-modules (with Merlin)
- 1986: protocol derivation (with Gotzhein)
- 2002 with Yamaguchi, El Fakih (Osaka, Japon)
- 2006 with Braek, Castejon (Trondheim, Norway)



Recurring research themes :

Protocol derivation from a given service specification (2)

Research issues:

- The importance of communication service specifications
  - my paper in IEEE Tr. Comm. of 1980 - “The Importance of the Service Concept in the Design of Data Communications Protocols” by Vissers et Logrippo in 1986
- Derivation algorithms
  - Original approach with Gotzhein in 1986
  - Generalization: allowing for weak sequencing (2008)
- Tools and automation
  - Tools for the case that the service specification is given in the form of a Petri net (research group at the University of Osaka – in the 1990ies)
  - Extended tool for the optimization of the allocation of resources to the different distributed system components (with El Fakih and Yamaguchi – 2002)
  - Tool for the case that the service specification is given in the form of a UML Activity diagram (Lamaarti, UofO, 2010)
- Use in the context of Web Services
  - Experimentation and demonstration: Service in the form of Activity diagram, generation of components (as Activity diagrams), automatic translation into BPEL processes, and implementation in the form of a distributed application using the SOAP protocol of Web Services.

Comments on collaborations

Collaborations have always been very important in my research

Longer visits

- Visiting professor 1977 at EPFL, Lausanne (Switzerland)
- Sabbatical year 1979 at Stanford University and SRI International (USA)
- Sabbatical year 1986 at Siemens in Munich (Germany)

Shorter visits

- 1980ies: CNET Lannion (France); Nokia (Finland); Grenoble (France); HMI Berlin; NTT Tokyo; Tsinghua University, Beijing
- 1990ies: University of Munich
- 2000s: University of Osaka (Japan); University of Trondheim (Norway); HNUST Xiangtan (China)

Ad hoc collaborations with

- researchers from other institutions: e.g. Merlin, Raynal, Gotzhein, Verjus, Higashino
- colleagues from University of Montreal: e.g. Armstrong, Lecarme, Cerny, Keller
- colleagues from University of Ottawa: e.g. Georganas, Amyot, Jourdan, Hall, Mouftah

Many long-term visitors from various countries, e.g. Germany, France, Brazil, Japan, China

Comments on teaching

The life of scientific and engineering concepts

- *some have short life and disappear*
- **others have permanent value and remain important for long time**
  - **These are the concepts important to teach**
  - Example: Object-oriented design : Simula 1967, Smalltalk 1980, Java 1995, Web Services (they are nothing but OO and message coding using XML)
  - Examples related to my research:
    - Behavior derivation for sub-modules – for different specification languages and for different applications
    - Protocol derivation from a given service specification – same principles for different specification languages
  - Important: **to teach the concepts that remain valid / applicable for long time (while the technologies change)**

Influence of my research experience on teaching

- I know very well the domain in which I have done research
  - I feel confident to teach such subjects
  - To mention my old publications related to the subject makes my teaching more credible
- Some aspects of my current research are discussed in my graduate courses - learning how to do research: critical thinking - how to find a good research topic - methods and tools for problem solving - experimenting with examples

Concluding remarks

The title: Research Issues in the Areas of Programming Languages and Communication Protocols: My Personal Experience since the 1970ies

Question: What is the impact today ?

Answer: There are several examples of advances in computer science and engineering that are related to my work on communication protocols:

- The architecture of communication protocols in layers
- Model checking for distributed systems
- Several notations of UML and related tools
- Model-based system development

Impact today

The layered architecture of communication protocols

- These concepts are generally accepted and used for the design of networks and distributed systems

Model checking for distributed systems

- Today’s tools for model checking of distributed systems are based on reachability analysis and the earlier tools for verifying the absence of deadlocks and non-specified receptions in communication protocols; they offer in addition the possibility of verifying that specific properties, specified in temporal logic, are satisfied. An example is the SPIN tool.

Impact today (2)

Notations of UML and related tools

- Among the three FDTs (Estelle, LOTOS et SDL), SDL was the most successful. It has been used for describing many standard communication protocols and other industrial systems, and the commercial tools have been used by industry, for example to build wireless telephony systems. Recently, SDL was integrated as a profile into UML-2, and the tools have been adapted to this new context.

Model-based system development

- Model-based system development has become mainstream in software engineering. In the context of protocol engineering, this approach has been used from the beginning. The specification of the protocol is an abstract model of all the implementations of that protocol, and the verification of the protocol is done at this abstract level. In fact, the FDTs SDL and Estelle, as also Harel's State Charts of 1987, are based on the concept of extended state machines from the 1970ies, and they can be considered the ancestors of today's State Machine diagrams of UML.

A final remark

Looking back on a life-long research career, an important aspect are the many friendships that have developed over the years.

At this occasion, I would like to thank all the people I have met for their welcome and - in some cases - the exciting collaborations that emerged.

Questions ??

Comments ??

These slides can be downloaded from

http://www.site.uottawa.ca/~bochmann/talks/ResearchIssues.ppt