Topics in Software Engineering

Assoc-Prof John Grundy
Dept of Computer Science
University of Auckland

Outline

- What is Software Engineering?
- Some representative "topics" in Software Engineering
  - Overview of general area of research+practice
  - Examples of my research work contributions
- The future of Software Engineering
  - Research/practice directions
  - Implications for SE Education

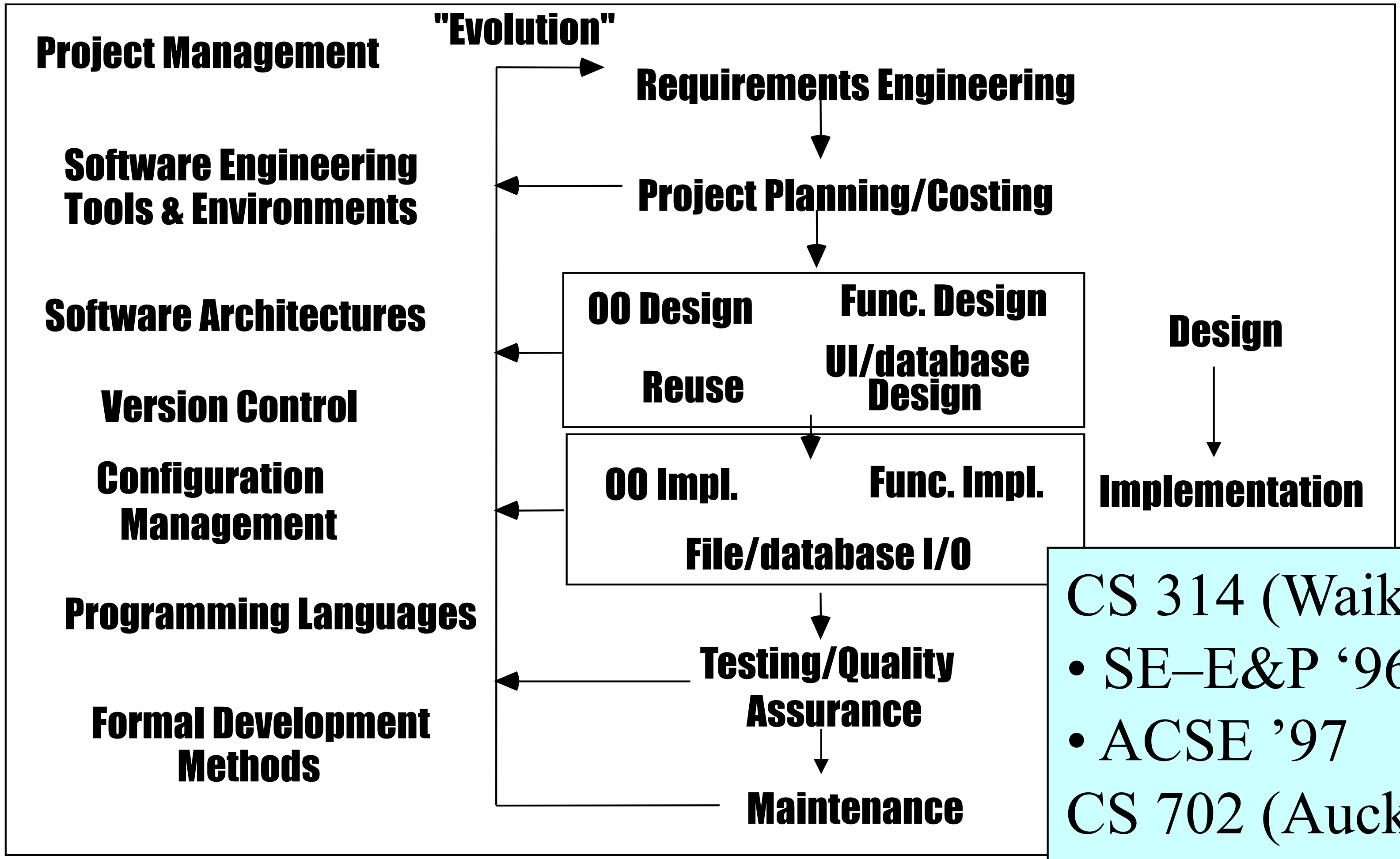
Topics from:
COMPSCI 702
- JIE&R 2000
- ACSE '97

What Software Engineering is (and isn't!)...

- Disciplined application of formal, semi-formal and informal techniques to make large-scale software development more effective and efficient
- Theory + Practical Techniques + Experience
- What its NOT:
  - Programming
  - "Formal methods"
  - Product Design
  - Quality assurance
- All are PART of SE, but SE is much more than this...

Some Key Aspects of Software Engineering…

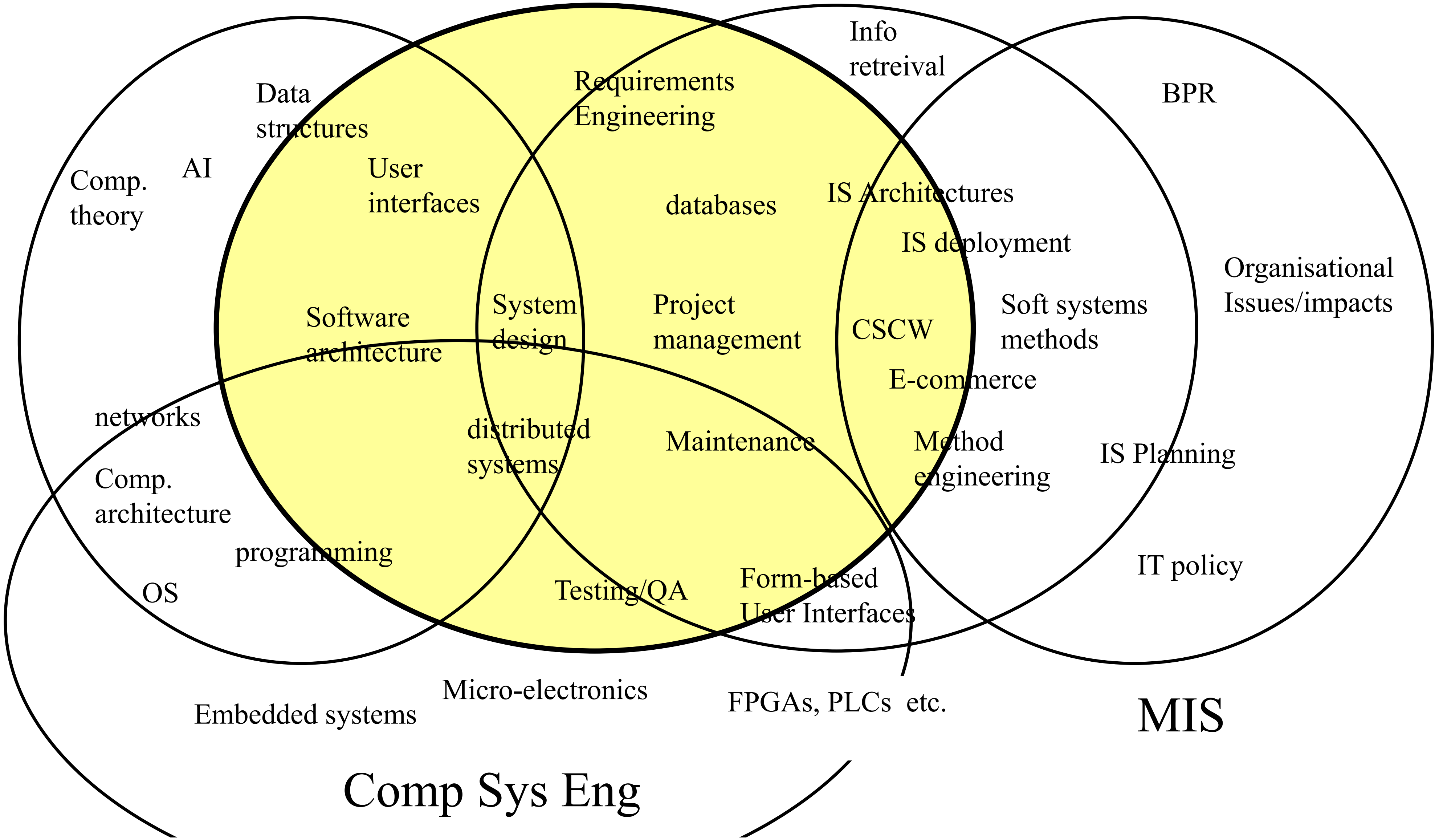
Software Processes



CS 314 (Waikato):
- SE–E&P '96
- ACSE '97

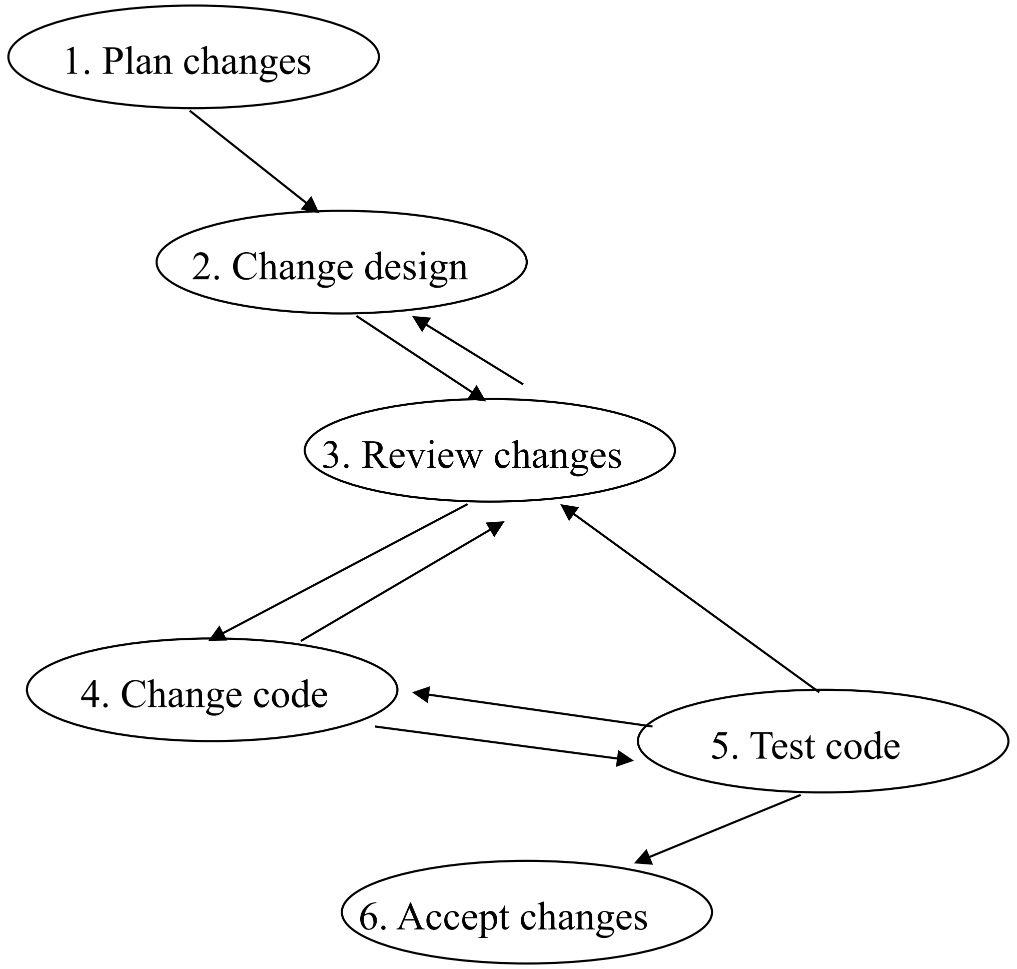
CS 702 (Auckland):
- JIE&R 2000

CS vs SE vs ISE vs MIS...

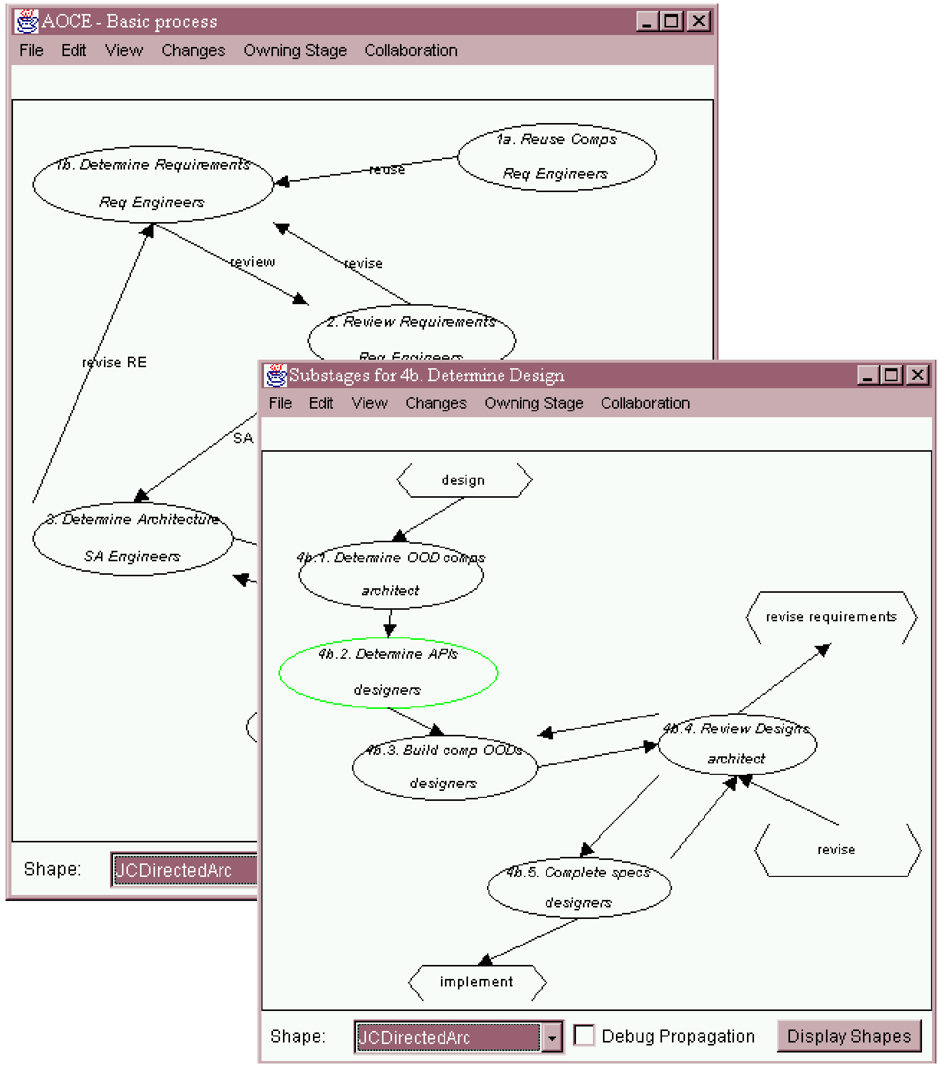


1. Software Processes

- "Steps" used to build software
- "Stages" in process
- Codification of life-cycle steps...
- "Waterfall" vs "Spiral" vs "RAD" vs "XP" ...
- **Good process = good product!**



Example: Serendipity-II



- Model SE processes, roles, resources, instantiate work plans
- Enact ("run") processes
- Collaborative modelling
- Process/plan templates
- Task automation/integration agents
- Co-ordinate use of other tools…

Serendipity, Serendipity-II:
- ASE 1998, IEEE IC 1998
- IJSEKE 1999, JEUC 1998

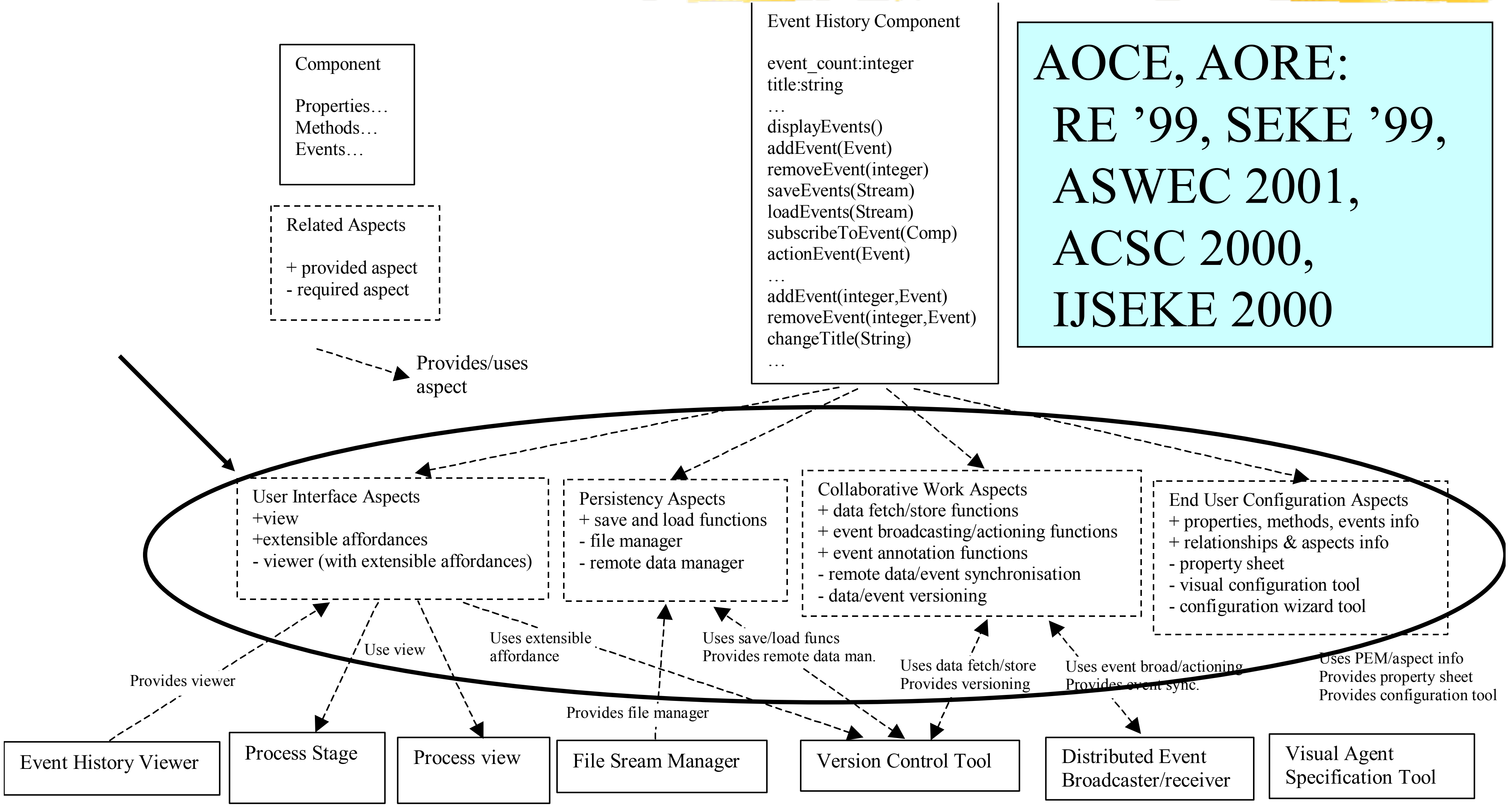
2. Requirements Engineering

- Determine what software should do

  (i.e. **user** requirements) – actually very hard!!
- Codify user requirements e.g. “use cases”, “structured English”, “goal-directed reqs”
- Specify system e.g. Object-oriented & Structured Analysis
- “Non-functional” constraints – performance, quality of service, interfaces

Question - What’s the more important:
“Do the right thing” vs “Do the thing right” ??

Example: Aspect-Oriented Requirements Engineering

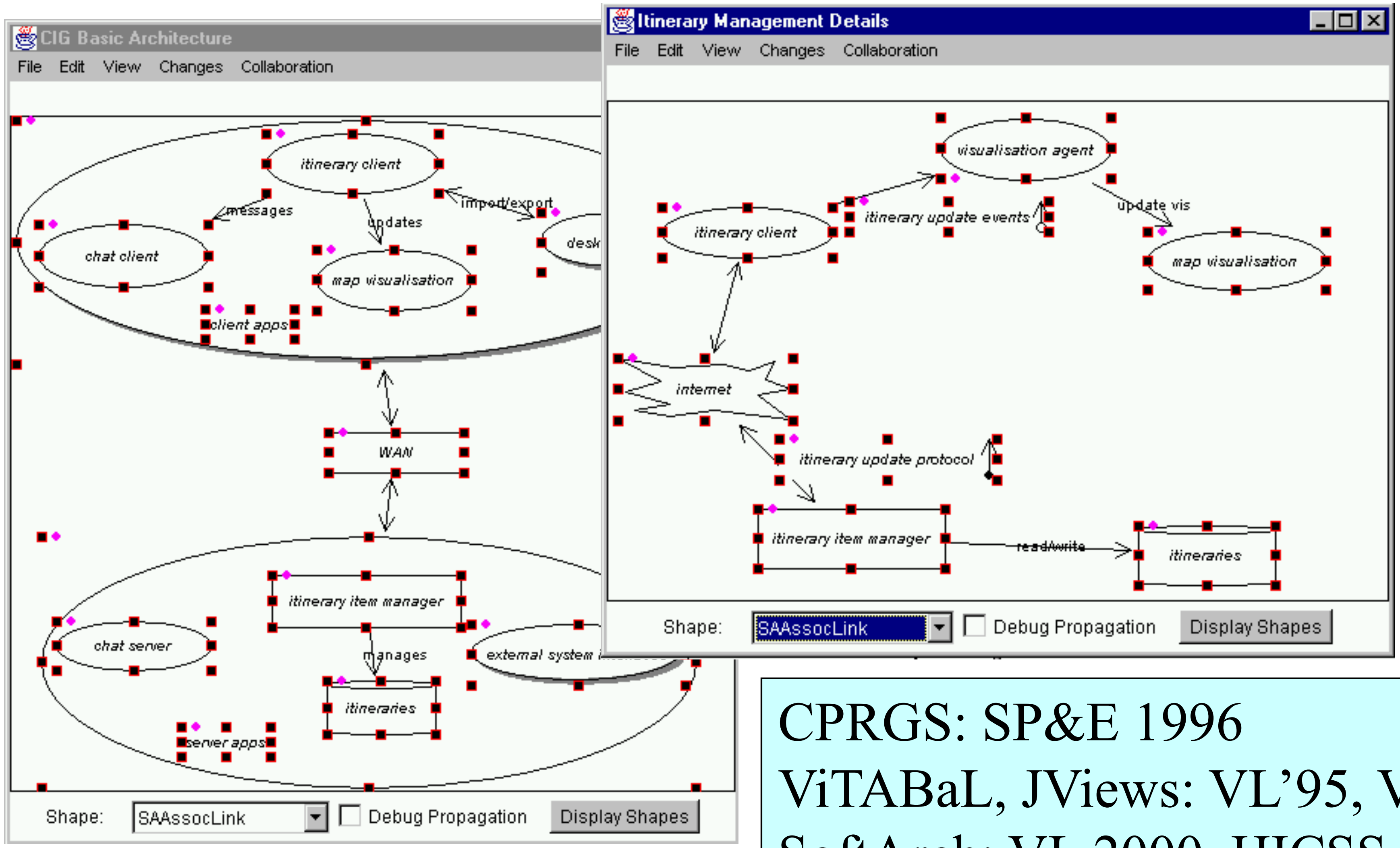


AOCE, AORE:
RE '99, SEKE '99,
ASWEC 2001,
ACSC 2000,
IJSEKE 2000

3. Software Architectures

- c.f. hardware architecture (bus, CPU, memory etc)
- Organisation of overall system
- Machines to run software on
- Networks to connect machines
- Allocation of processes to machines ("programs")
- Communication between processes/programs
- Synchronization of processing/data update
- "Patterns" of common architectural solutions e.g. client-server

Examples: SoftArch, CPRGs, ViTABaL, JViews, …

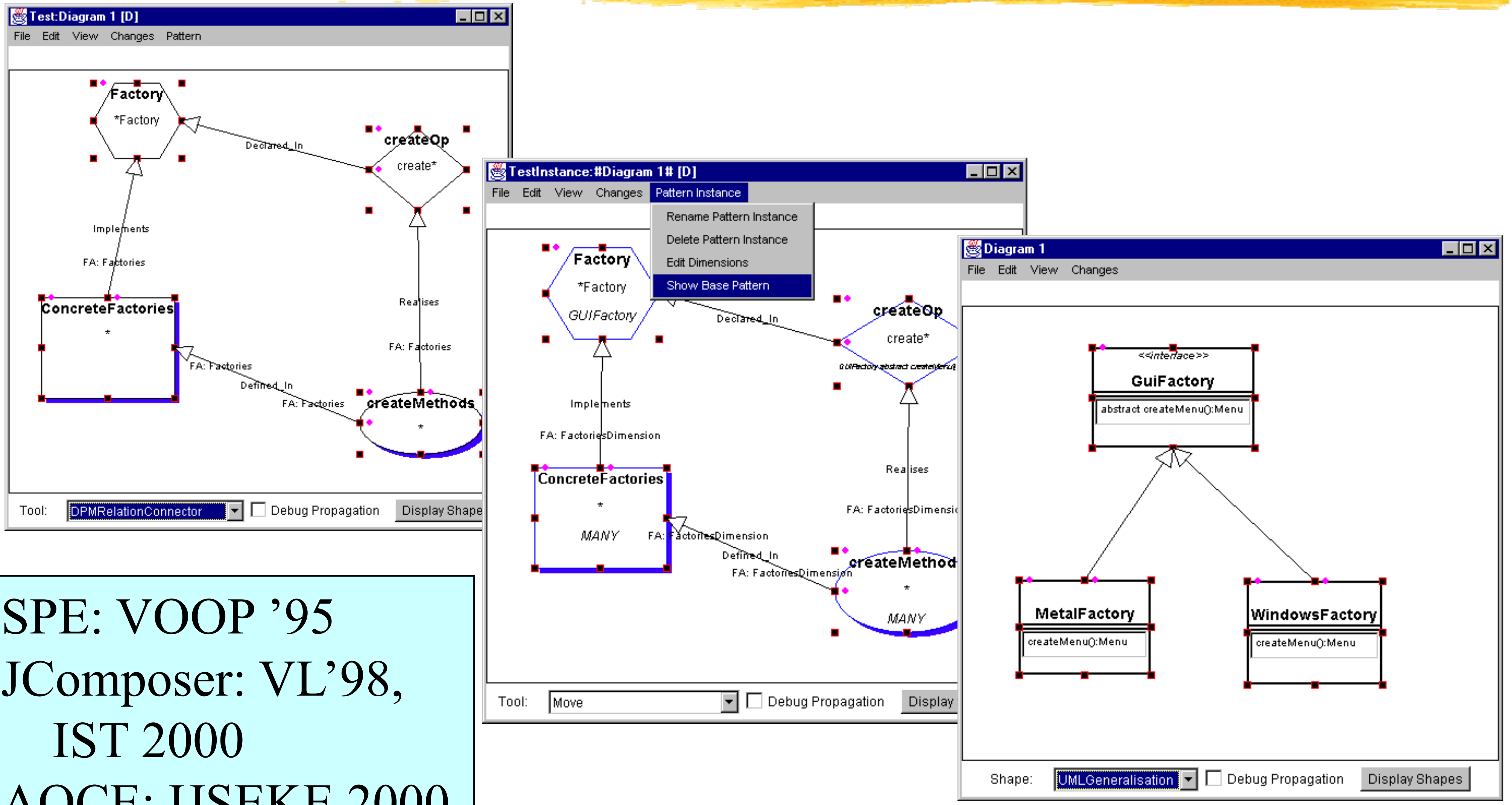


CPRGS: SP&E 1996
ViTABaL, JViews: VL'95, VL'98
SoftArch: VL 2000, HICSS 2000

4. System Design

- Refine specification to a detailed implementation description
- Program, user interface, database, algorithm design
- Specify designs using various formal/semi-formal/informal techniques
- Often, like Software Architectures, requires a great degree of creativity…

Examples: AOCE, SPE, OO/ER, JComposer, <u>DPTool</u>



SPE: VOOP ’95

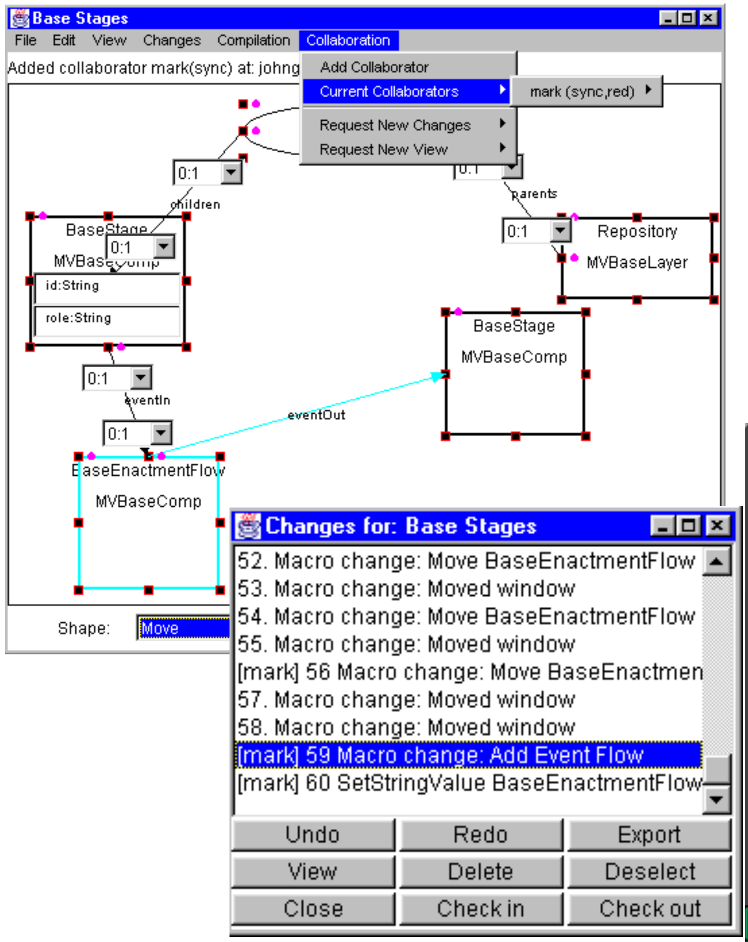
JComposer: VL’98,
IST 2000

AOCE: IJSEKE 2000

5. User Interfaces

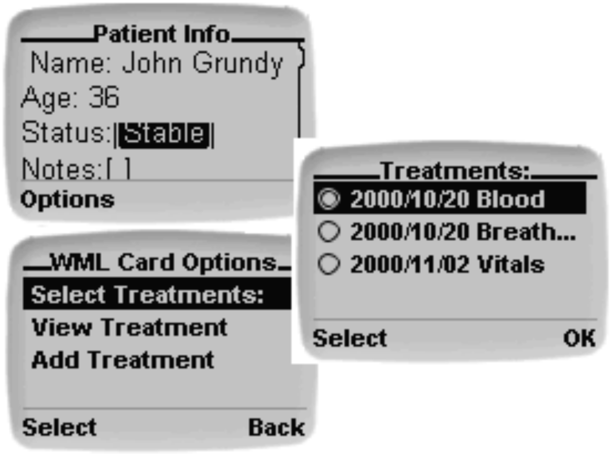
- Things that end-users of software interact with
- Huge range: desktop, web, hand-held (phone, PDAs), Virtual Reality, Ubiquitous
- Various applications: data entry, data visualisation, document creation, groupware, games etc
- Requires creative effort to design, build
- Very often a make-or-break part of the system

Examples: BBW, Skin, JViews, CPRGs, Groupware



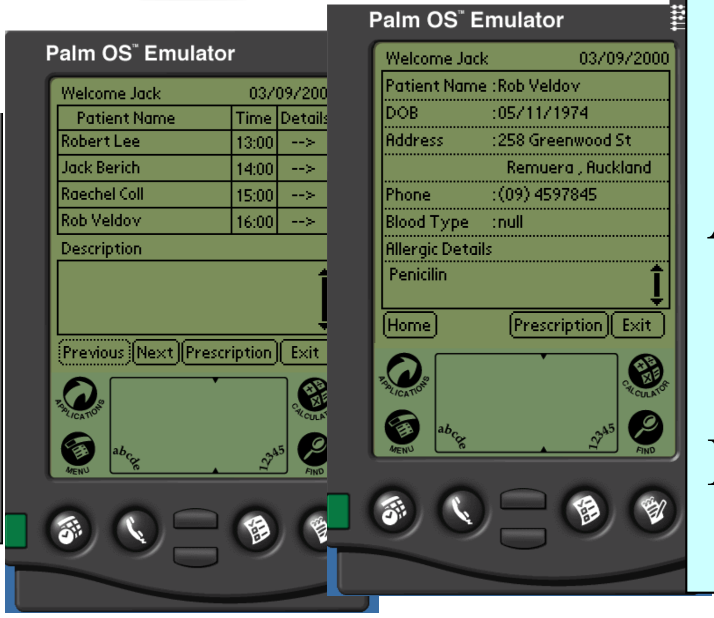
Added collaborator mark(sync) at: johng

```
52. Macro change: Move BaseEnactmentFlow
53. Macro change: Moved window
54. Macro change: Move BaseEnactmentFlow
55. Macro change: Moved window
[mark] 56 Macro change: Move BaseEnactmen
57. Macro change: Moved window
58. Macro change: Moved window
[mark] 59 Macro change: Add Event Flow
[mark] 60 SetStringValue BaseEnactmentFlow
```



Name: John Grundy
Age: 36
Status: Stable
Notes: [ ]

- 2000/10/20 Blood
- 2000/10/20 Breath...
- 2000/11/02 Vitals



Welcome Jack 03/09/200

Patient Name	Time	Details
Robert Lee	13:00	-->
Jack Berich	14:00	-->
Raechel Coll	15:00	-->
Rob Veldov	16:00	-->

Description

Welcome Jack 03/09/2000

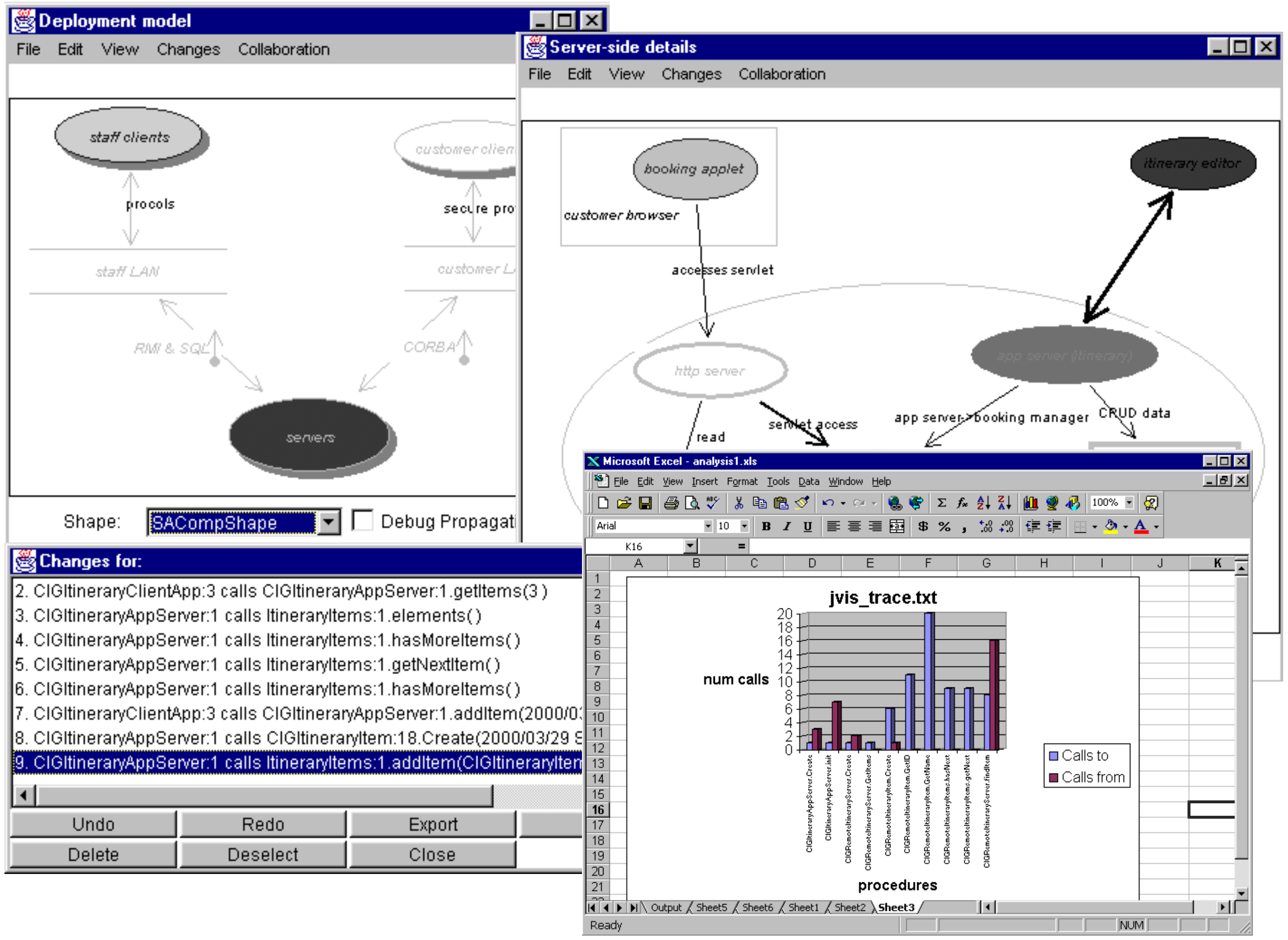
Patient Name	:Rob Veldov
DOB	:05/11/1974
Address	:258 Greenwood St
	Remuera , Auckland
Phone	:(09) 4597845
Blood Type	:null
Allergic Details	
Penicilin	

Groupware:
  OZCHI 95, 96;
  APCHI '98;
  EHCI '98
BBW:
  OZCHI '98;
  IST 2000
Adpatable UIs:
  AUIC 2000,
  IwC 2001
Mobile UIs:
  OOIS 2001

6. Quality Assurance

- How do we know our software is any good???
- Usability – evaluating user interfaces e.g. observation, interviews, questionaires
- Conformance – black-box/white-box unit tests
- Performance – benchmarking tests
- Organisational effectiveness – cost-benefit analysis, organisational impact, ethnography

Examples: SoftArch, ViTABaL, SPE, CernoII, ...



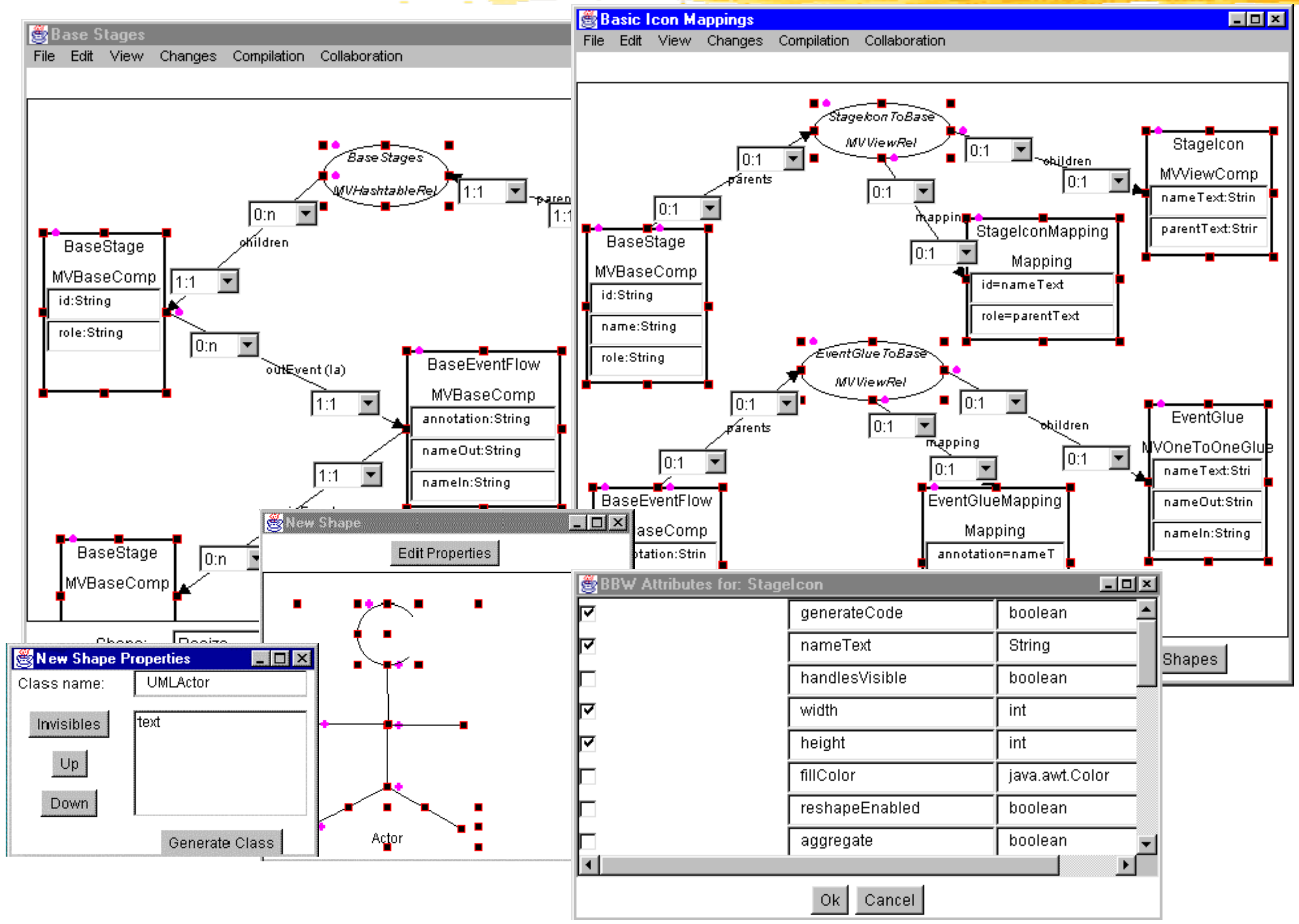
2. CIGItineraryClientApp:3 calls CIGItineraryAppServer:1.getItems(3)
3. CIGItineraryAppServer:1 calls ItineraryItems:1.elements()
4. CIGItineraryAppServer:1 calls ItineraryItems:1.hasMoreItems()
5. CIGItineraryAppServer:1 calls ItineraryItems:1.getNextItem()
6. CIGItineraryAppServer:1 calls ItineraryItems:1.hasMoreItems()
7. CIGItineraryClientApp:3 calls CIGItineraryAppServer:1.addItem(2000/03
8. CIGItineraryAppServer:1 calls CIGItineraryItem:18.Create(2000/03/29
9. CIGItineraryAppServer:1 calls ItineraryItems:1.addItem(CIGItineraryItem

ViTABaL:
VL '95,
IWSA '96
Cerno-II/SPE:
VOOP '95
SoftArch:
CoSET 2000,
SM&T 2000,
VL 2000

7. Software Engineering Tools

- Systems are now so large, complex that NEED good tools to effectively develop
- Huge range: process/project management; requirements & analysis; formal specification/refinement; design; implementation; testing/QA; maintenance; documentation; deployment; "CASE"
- Building software tools HARD – meta-CASE

Ispel, SPE, ViTABaL, Serendipity, JComposer/BBW



New Shape Properties

Class name: UMLActor

text

Invisibles | Up | Down | Generate Class

BBW Attributes for: StageIcon

☑	generateCode	boolean
☑	nameText	String
☐	handlesVisible	boolean
☑	width	int
☑	height	int
☐	fillColor	java.awt.Color
☐	reshapeEnabled	boolean
☐	aggregate	boolean

Ok | Cancel

NZJC '93, '95;
VL'93, '98
VOOP '95;
SEE '95;
ER '95;
APSEC '95
S-P&E '96;
IEEE TOSE'98;
IJSEKE '99;
IST 2000;
ASWEC 2001;
…

The Future of Software Engineering??

- We survived Y2K, so now what???
- Larger, more de-centralised systems
- System integration, including legacy systems
- Ubiquitous & hand-held computing growth
- More packaged abstractions (reusable software "components")
- Virtual software teams; end-user computing
- Tools – complexity management
- Quality assurance => how get better software…

Implications for SE Education...

- New processes/methods/tools: XP; RAD; goal-directed RE; component-based software development; new languages/IDEs; new CASE
- "New" culture of assemble/tailor vs build
- Awareness of standardisation efforts e.g. XML
- Certification of software engineers?

- Work with real users; with real developers
- Solid understanding of concepts (theory) + best-practice techniques + experience
- Continuous up-skilling

JIE&R 2000
ACSE '97
ASAW 2000

Publications (1)

- Grundy, J.C., Mugridge, W.B. and Hosking, J.G. Constructing component-based software engineering environments: issues and experiences, Information and Software Technology Vol 42, No. 2, Special Issue on Constructing Software Engineering Tools, Elsevier Science Publishers
- Grundy, J.C. Multi-perspective specification, design and implementation of components using aspects, International Journal of Software Engineering and Knowledge Engineering, Vol. 10, No. 6, December 2000, World Scientific
- Grundy, J.C., Hosking, J.G., Mugridge, W.B., Apperley, M.D. A decentralised architecture for software process modelling and enactment, IEEE Internet Computing: Special Issue on Software Engineering via the Internet, Vol. 2, No. 5, September/October 1998, IEEE CS Press, pp. 53-62
- Grundy, J.C., Hosking, J.G., Mugridge, W.B. Inconsistency Management for Multi-view Software Development Environments, IEEE Transactions on Software Engineering: Special Issue on Managing Inconsistency in Software Development, Vol. 24, No. 11, 1998, IEEE CS Press
- Grundy, J.C., and Hosking, J.G., Mugridge, W.B., Supporting flexible consistency management via discrete change description propagation, Software - Practice and Experience, Vol. 26, No. 9, September 1996, Wiley, 1053-1083
- Grundy, J.C. and Hosking, J.G. Serendipity: integrated environment support for process modelling, enactment and work coordination, Automated Software Engineering: Special Issue on Process Technology, Vol. 5, No. 1, January 1998, Kluwer Academic Publishers, pp. 27-60
- Grundy, J.C., Mugridge, W.B., Hosking J.G. Supporting Large-scale End-user specification of workflows, work coordination and tool integration, Journal of End-User Computing, Vol. 10, No. 2, May 1998, Idea Group Publishing, pp. 39-49.
- Grundy, J.C., and Hosking, J.G., Constructing integrated software development environments with MViews, International Journal of Applied Software Technology 2 (3-4), 133-160

Publications (2)

- Grundy, J.C., Cai, Y. and Liu, A. Generation of Distributed System Test-beds from High-level Software Architecture Descriptions, In Proceedings of the 16th International Conference on Automated Software Engineering, San Diego, 26-29 Nov 2001, IEEE CS Press, pp. 193-200.
- Grundy, J.C. and Hosking, J.G. Engineering plug-in software components to support collaborative work, Software - Practice and Experience, Vol. 32, No. 10, August 2002, Wiley, 983-1013
- Grundy, J.C. and Hosking, J.G. Developing Adaptable User Interfaces for Component-based Systems, Interacting with Computers, vol. 14, no. 3, March 2002, Elsevier, pp. 175-194
- Olsen, T. and Grundy, J.C. Supporting traceability and inconsistency management between software artefacts, In Proceedings of the 2002 International Conference on Software Engineering and Applications, Boston, MA, 2-5 Nov 2002
- Li, Y., Grundy, J.C., Amor, R. and Hosking, J.G. A data mapping specification environment using a concrete business form-based metaphor, In Proceedings of the 2002 International Conference on Human-Centric Computing, IEEE CS Press.
- Grundy, J.C. Aspect-oriented Requirements Engineering for Component-based Software Systems, 1999 IEEE Symposium on Requirements Engineering, Limmerick, Ireland, 7-11 June, 1999, IEEE CS Press
- Grundy, J.C., Mugridge, W.B. and Hosking, J.G. Visual specification of multiple view visual environments, In Proceedings of IEEE VL'98, Halifax, Nova Scotia, Canada, Sept 1-4, 1998, IEEE CS Press, pp. 236-243.
- Grundy, J.C. Storage and retrieval of Software Components using Aspects, In Proceedings of the 2000 Australasian Computer Science Conference, Canberra, Australia, Jan 30-Feb 3 2000, IEEE CS Press, pp 95-103
- Grundy, J.C., Mugridge, W.B., Hosking, J.G. and Kendal, P. Generating EDI Message Translations from Visual Specifications, In Proceedings of the 16th International Conference on Automated Software Engineering, San Diego, 26-29 Nov 2001, IEEE CS Press, pp. 35-42.
- Grundy, J.C., Hosking, J.G., Software environment support for integrated formal program specification and development, In Proceedings of the 1995 Asia-Pacific Software Engineering Conference, Brisbane, December 1995, IEEE CS Press, pp. 264-273.

Publications (3)

- Grundy, J.C. Teaching Information Systems by simulations: issues and experience, Journal of Informatics - Education and Research, vol. 4, no. 1, Spring 2002, IAIM Press, pp. 47-58.
- Grundy, J.C. A Graduate Course on E-commerce Information Systems Engineering, Journal of Informatics Education and Research, IAIM Press, Vol. 2, No. 2.
- Grundy, J.C. and Liu, A. Directions in Engineering Non-Functional Requirement Compliant Middleware Applications, In proceedings of the 3rd Australasian Workshop on Software and System Architectures, Sydney, Australia, 19-20 Nov, 2000
- Grundy, J.C. Experiences in Facilitating Student Learning in a Group Information Systems Project Course, in Proceedings of the 1996 Software Engineering: Education and Practice Conference (SE:E+P'96), Dunedin, New Zealand, 1996, IEEE CS Press
- Grundy, J.C.Comparitive Analysis of Design Principles for Project-based IT Courses, in Proceedings of the 2nd Australasian Conference on Computer Science Education, Melbourne, Australia, 1997, ACM Press, pp. 170-177
- Grundy, J.C. Integrating software architecture topics into a software engineering curriculum, In Proceedings of the 1999 Australasian Workshop on Software Architecture, Sydney, Australia Nov 1999