Software Architecture Modelling, Analysis and Implementation with SoftArch

John Grundy
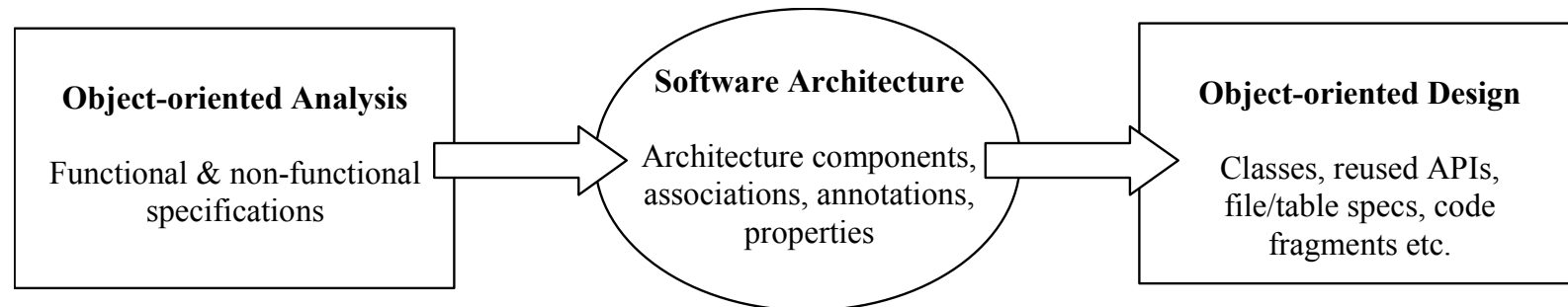
University of Auckland
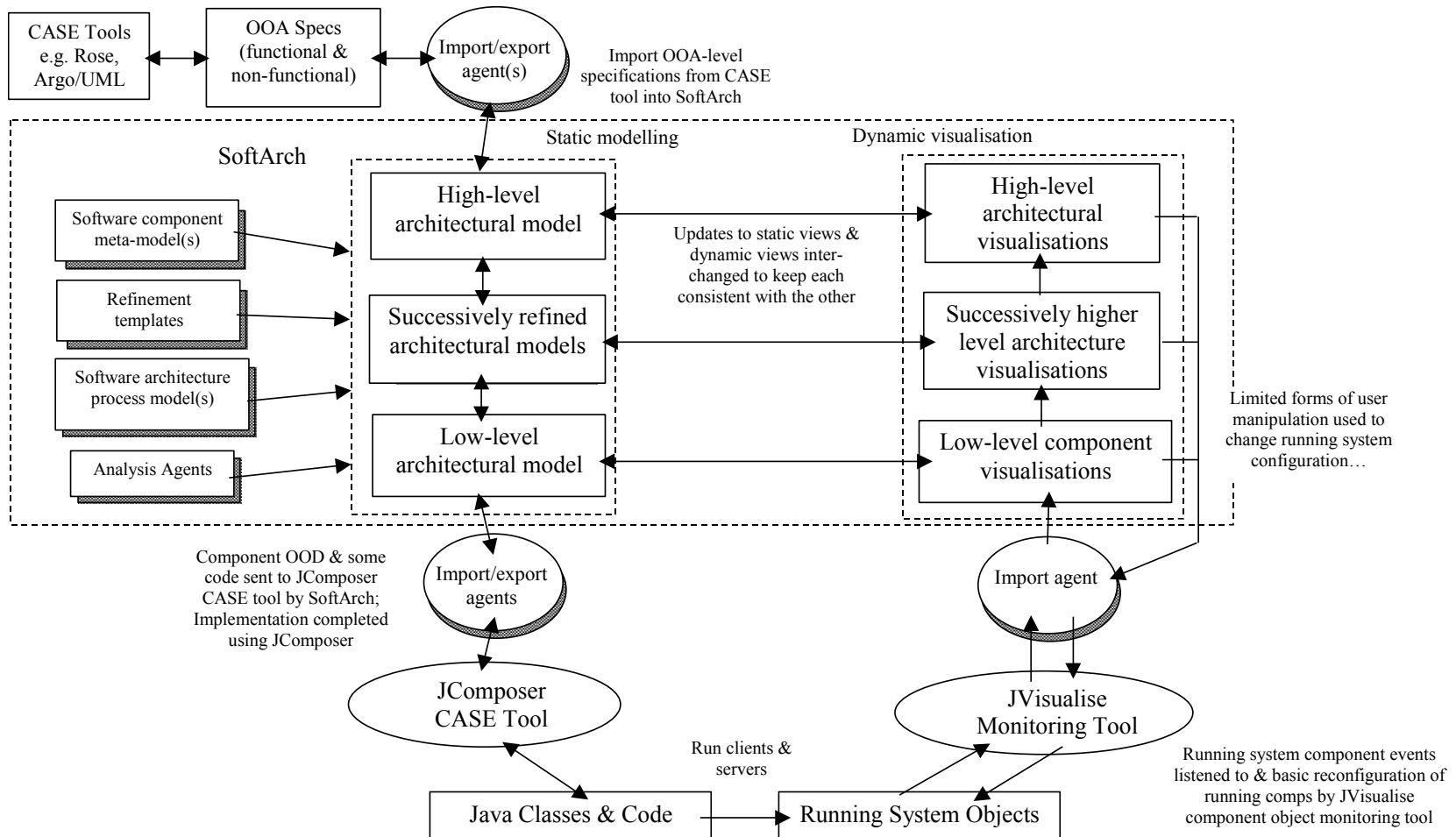
New Zealand

Overview

- Software Architecture vs OOA and OOD
- Modelling SAs in SoftArch:
  - Architectural abstractions
  - Refinement
  - Templates
- Architecture analysis
- Architecture visualisation
- SoftArch architecture…

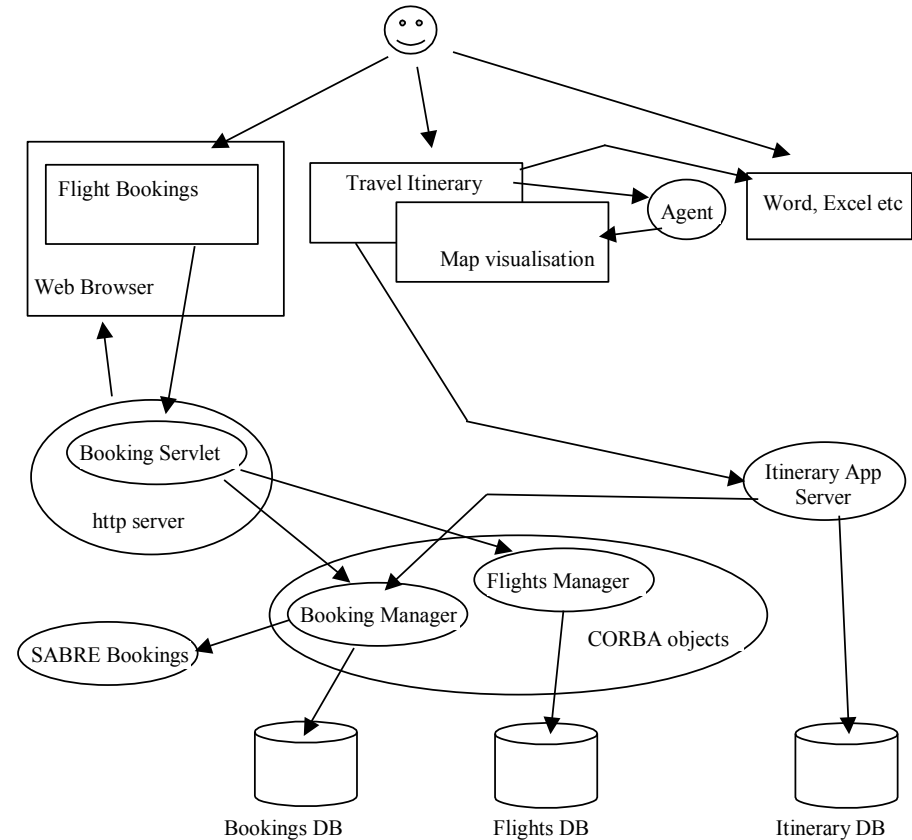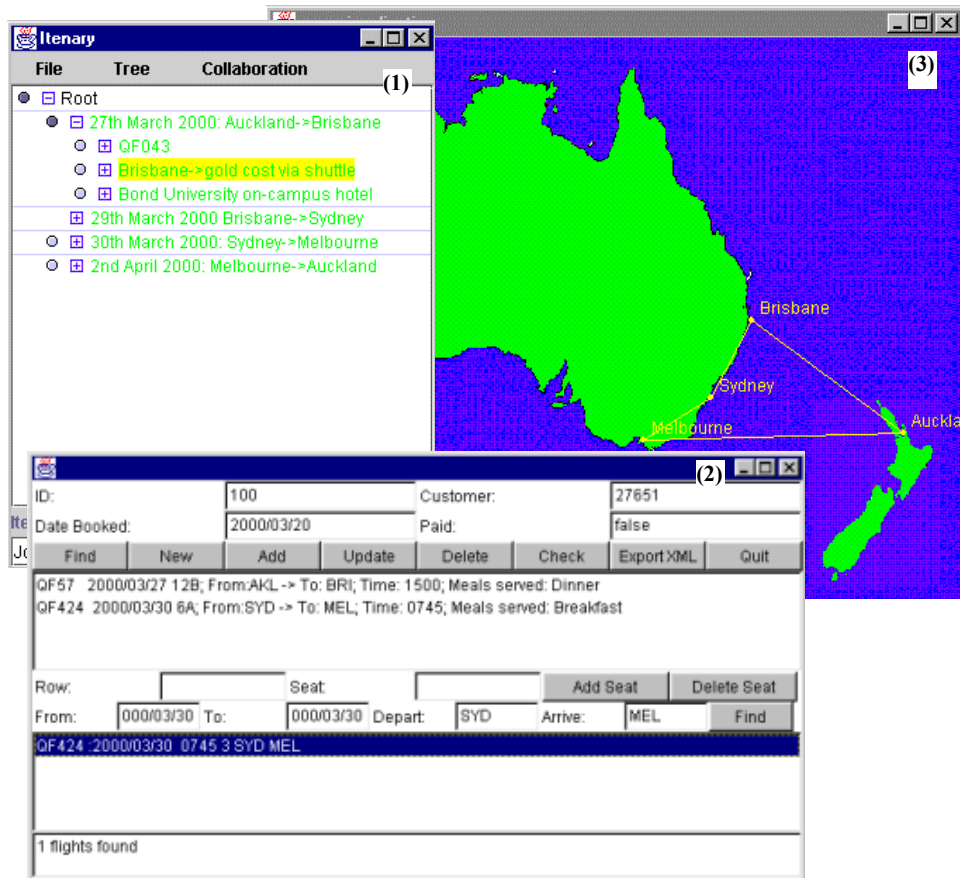# Software Architecture - our spin...

- Translation of OOA specification to OOD design via well-formed refinements
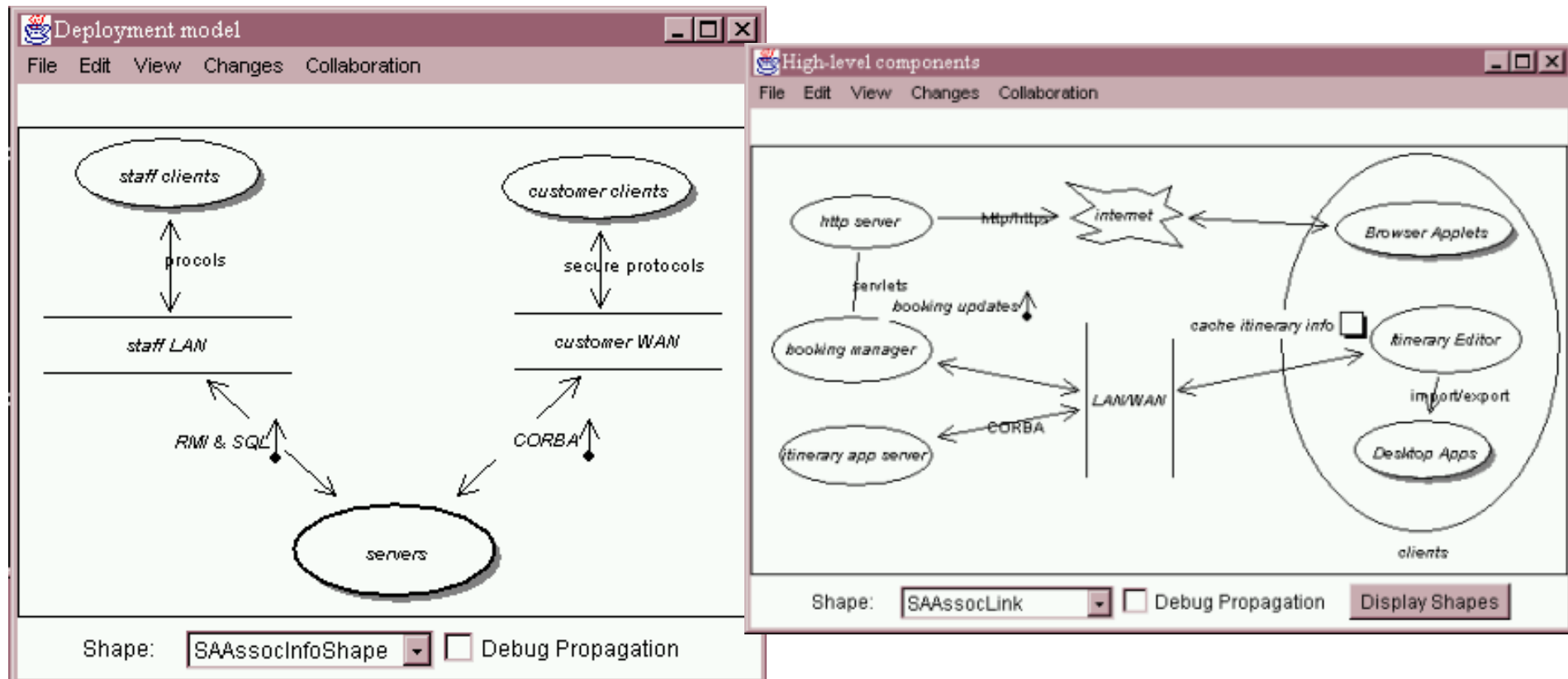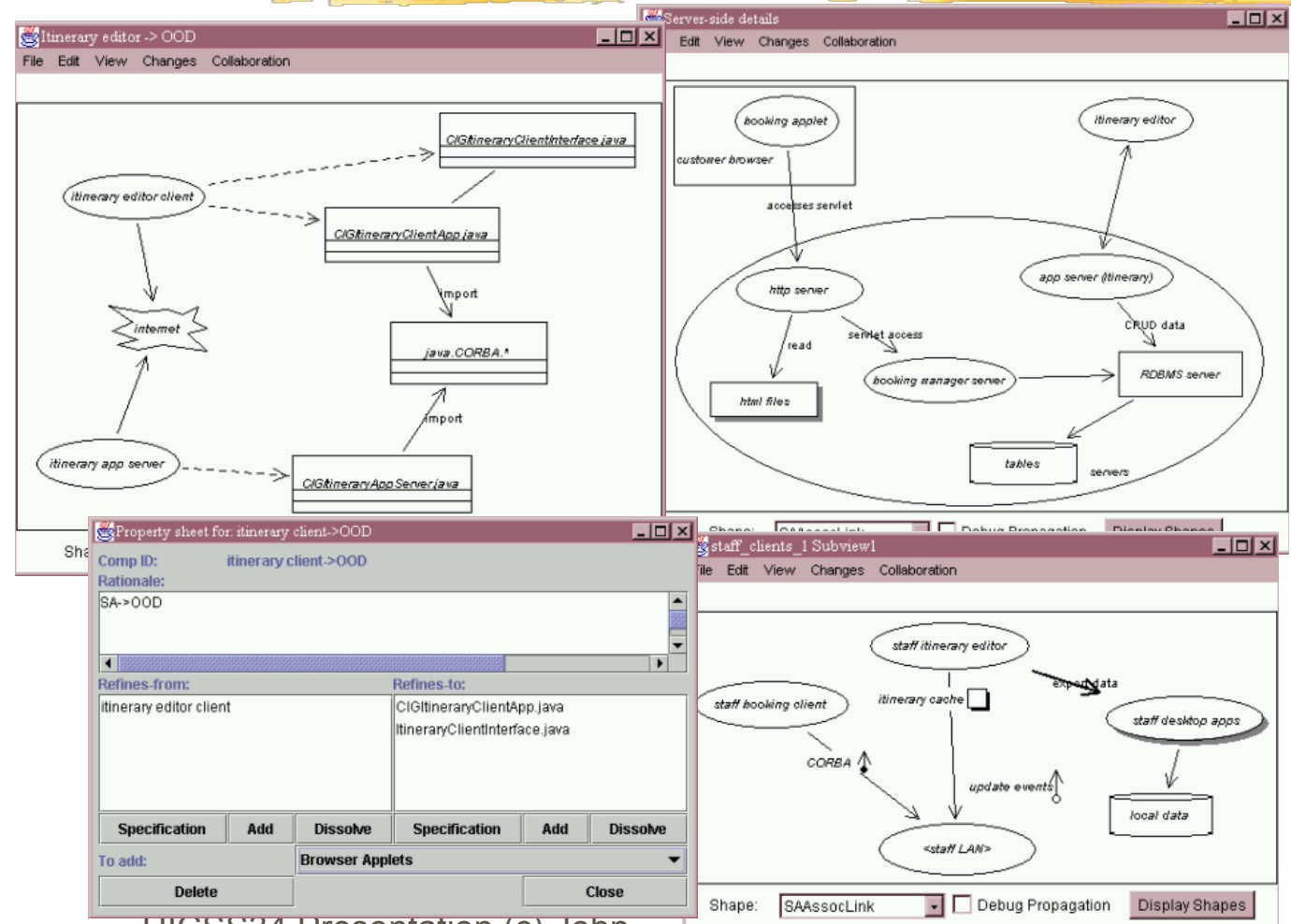- SA organises OOA for implementation, adhering to functional/non-functional specs

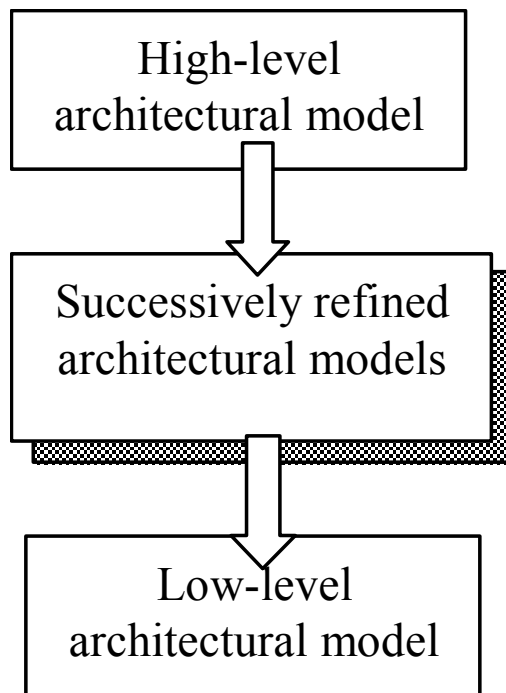| Object-oriented Analysis	Software Architecture	Object-oriented Design
Functional & non-functional specifications	Architecture components, associations, annotations, properties	Classes, reused APIs, file/table specs, code fragments etc.

Overview of SoftArch

CASE Tools e.g. Rose, Argo/UML ↔ **OOA Specs** (functional & non-functional) ↔ **Import/export agent(s)**

Import OOA-level specifications from CASE tool into SoftArch

SoftArch

Static modelling

Dynamic visualisation

- Software component meta-model(s)
- Refinement templates
- Software architecture process model(s)
- Analysis Agents

High-level architectural model

Successively refined architectural models

Low-level architectural model

Updates to static views & dynamic views inter-changed to keep each consistent with the other

High-level architectural visualisations

Successively higher level architecture visualisations

Low-level component visualisations

Limited forms of user manipulation used to change running system configuration…

Component OOD & some code sent to JComposer CASE tool by SoftArch; Implementation completed using JComposer

Import/export agents

JComposer CASE Tool

Java Classes & Code

Run clients & servers

Running System Objects

Import agent

JVisualise Monitoring Tool

Running system component events listened to & basic reconfiguration of running comps by JVisualise component object monitoring tool

HICSS34 Presentation (c) John Grundy 2000

Example System to Model...

Modelling...

Refinement

High-level architectural model

Successively refined architectural models

Low-level architectural model

Processes, templates, meta-modelling...
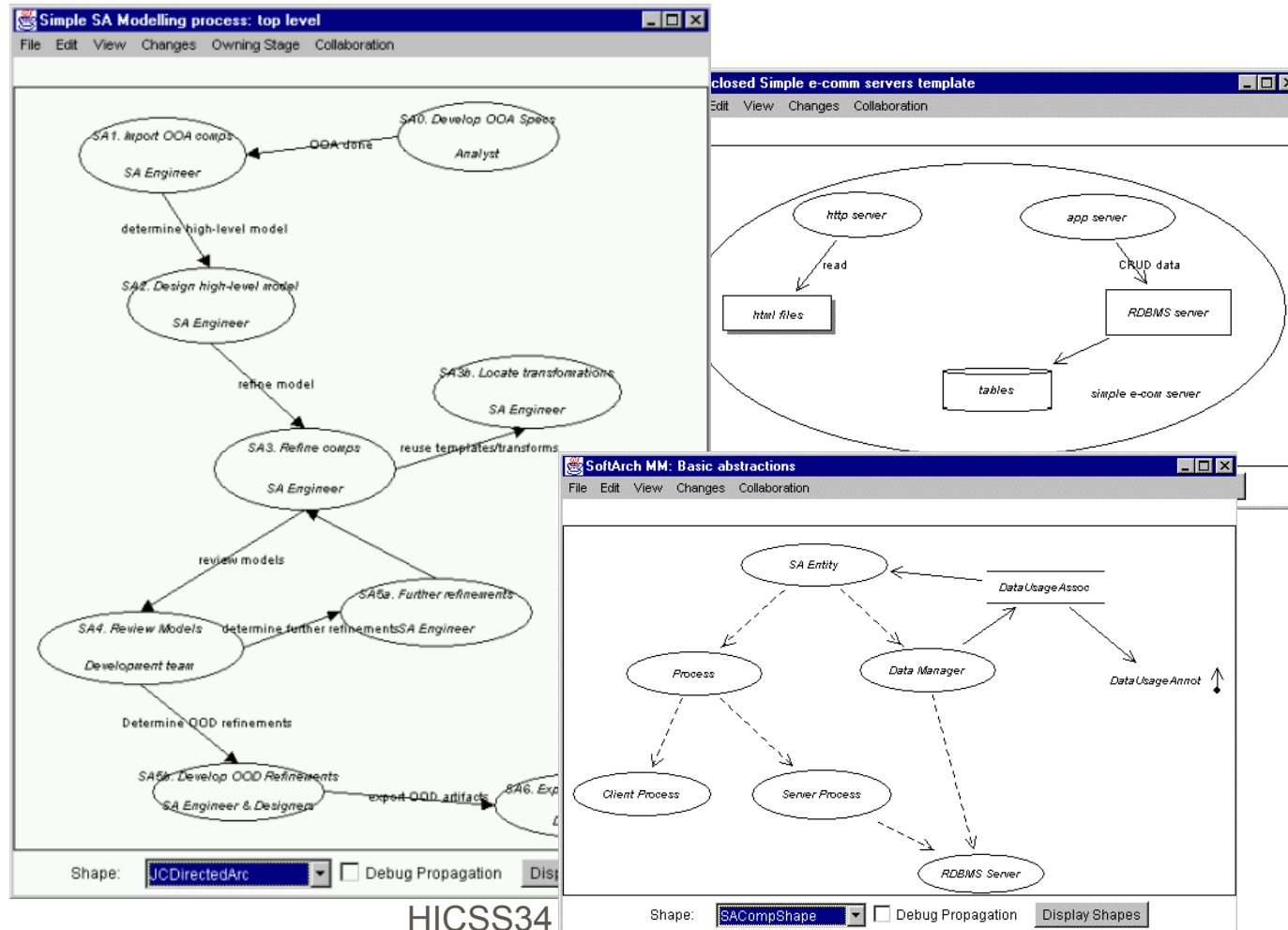
Grundy 2000

Architecture Analysis

Dynamic Visualisation

Other features...

- Import/export:
  - OOA spec from CASE tool
  - OOD to CASE tool

- Code generation:
  - Limited (currently via our JComposer CASE tool)
  - Planned: generate test bed for MTE project middleware analysis examples

- Analysis: MTE & other performance results used during modelling/analysis

SoftArch Architecture...
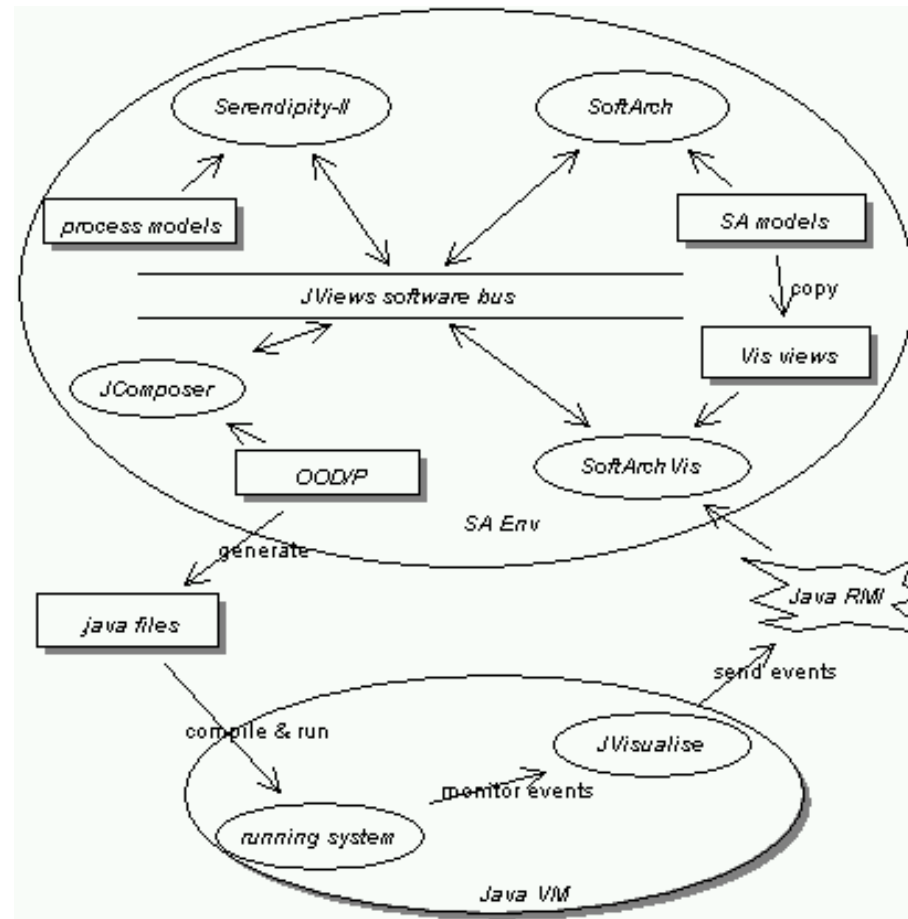
References

∎ Grundy, J.C. Software Architecture Modelling, Analysis and Implementation with SoftArch, In Proceedings of the 34th Hawaii International Conference on System Sciences (Software Architecture Mini-track), Jan 3-6, 2006, Maui, Hawaii, IEEE CS Press, pp. 3826 –3834.

∎ Grundy, J.C. and Hosking, J.G. SoftArch: tool support for integrated software architecture development, International Journal of Software Engineering and Knowledge Engineering, vol 13, no 2, April 2003, World Scientific, pp. 125-152.

∎ Stockle, H., Grundy, J.C. and Hosking, J.G. Notation Exchange Converters for Software Architecture Development, ESEC'2003 Workshop on Tool Integration, September 2003.

∎ Relating Software Requirements and Architectures, P. Avgeriou, J. Grundy, J. G. Hall, P. Lago, I. Mistrík (Eds), Springer , 2011, 430pp, ISBN 978-3-642-21000-6.

∎ Su, M.T., Hosking, J.G. and Grundy, J.C. KaitoroCap : a document navigation capture and visualisation tool, Demo Track of 2011 International Working Conference on Software Architecture (WICSA 2011), 21-24 June 2011, Boulder, Colorado.

∎ Wang, Y., Singh, S., Hosking, J.G. and Grundy, J.C. An Aspect-Oriented UML Tool for Software Development with Early Aspects, Proceedings of ICSE 2006 Workshop on Early Aspects at ICSE: Aspect-Oriented Requirements Engineering and Architecture Design, Shanghai, May 2006.

∎ Grundy, J.C., Cai, Y. and Liu, A. SoftArch/MTE: Generating Distributed System Test-beds from High-level Software Architecture Descriptions, Automated Software Engineering, Kluwer Academic Publishers, vol. 12, no. 1, January 2005, pp. 5-39.

∎ Cai, Y., Grundy, J.C., Hosking, J.G., Dai, X. Software Architecture Modelling and Performance Analysis with Argo/MTE, In Proceedings of the 2004 Conference on Software Engineering and Knowledge Engineering, Baniff, Canada, June 20-24 2004.

∎ Grundy, J.C., Wei, Z., Nicolescu, R. and Cai, Y. An Environment for Automated Performance Evaluation of J2EE and ASP.NET Thin-client Architectures, In Proceedings of the 2004 Australian Software Engineering Conference, Melbourne, Australia, April 14-17 2004, IEEE CS Press.

∎ 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. Directions in modelling large-scale software architectures, In Proceedings of the 2000 Conference on Software - Methods & Tools, Wollongong, Australia, Nov 6-10, 2000, IEEE CS Press.