A Service-Oriented Architecture for Software Process Technology

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Therese Helland
Computas AS
Norway
the@computas.com

John Grundy [1, 2] and John Hosking[1]
[1]Dept. of Computer Science and
[2]Dept. of Electrical and Computer Engineering
University of Auckland New Zealand
{john-g, john}@cs.auckland.ac.nz

Outline

- **Aims and requirements**
- **IMÅL approach and decentralised architecture**
- **PPML: Process modelling language**
- **Example usage**
  - Modelling and enactment
  - To do list
  - Process automation
- **Design & Implementation**
- **Evaluation**
- **Summary**

Aims and Requirements

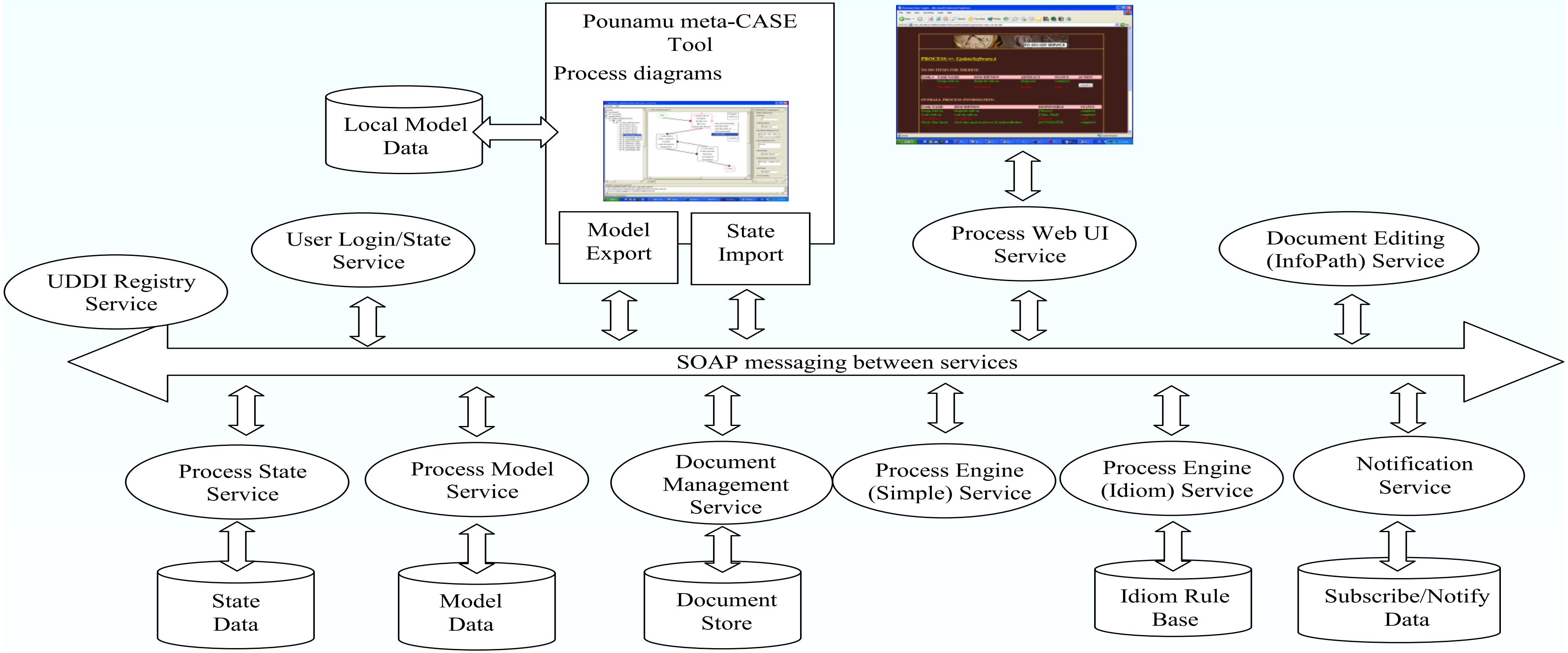
- **Aim:**
  - A tool suite providing a service oriented approach to software process modelling and enactment

- **Requirements:**
  - Configurable process modelling notation
  - Modelling and enactment for simple and complex process flow
  - Decentralised process management based on a set of pluggable services
  - Reuse COTS services as components

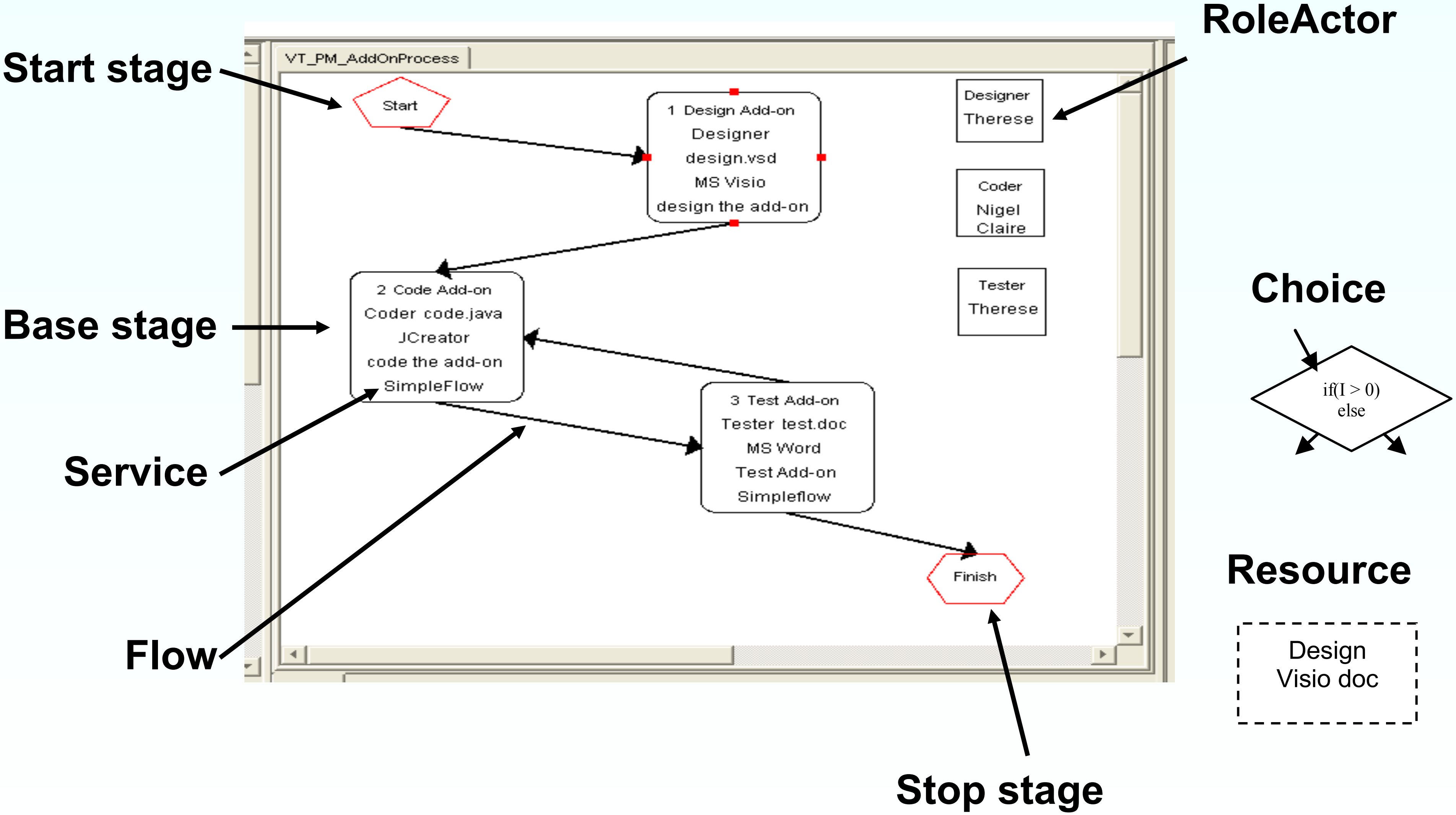
IMÅL approach

- **Simple and tailorable modelling language**
  - Has enactment visualisation capability
  - Based on our Serendipity language
- **Highly distributed service oriented process enactment engine**
  - Multiple process engine services
  - Process state, notification, login, to do list, etc services
- **3rd party component integration:**
  - InfoPath: document display
  - Idiom: rule-based decision making

IMÅL's decentralized Architecture



Process Modelling Notation: PPML



Start stage

Base stage

Service

Flow

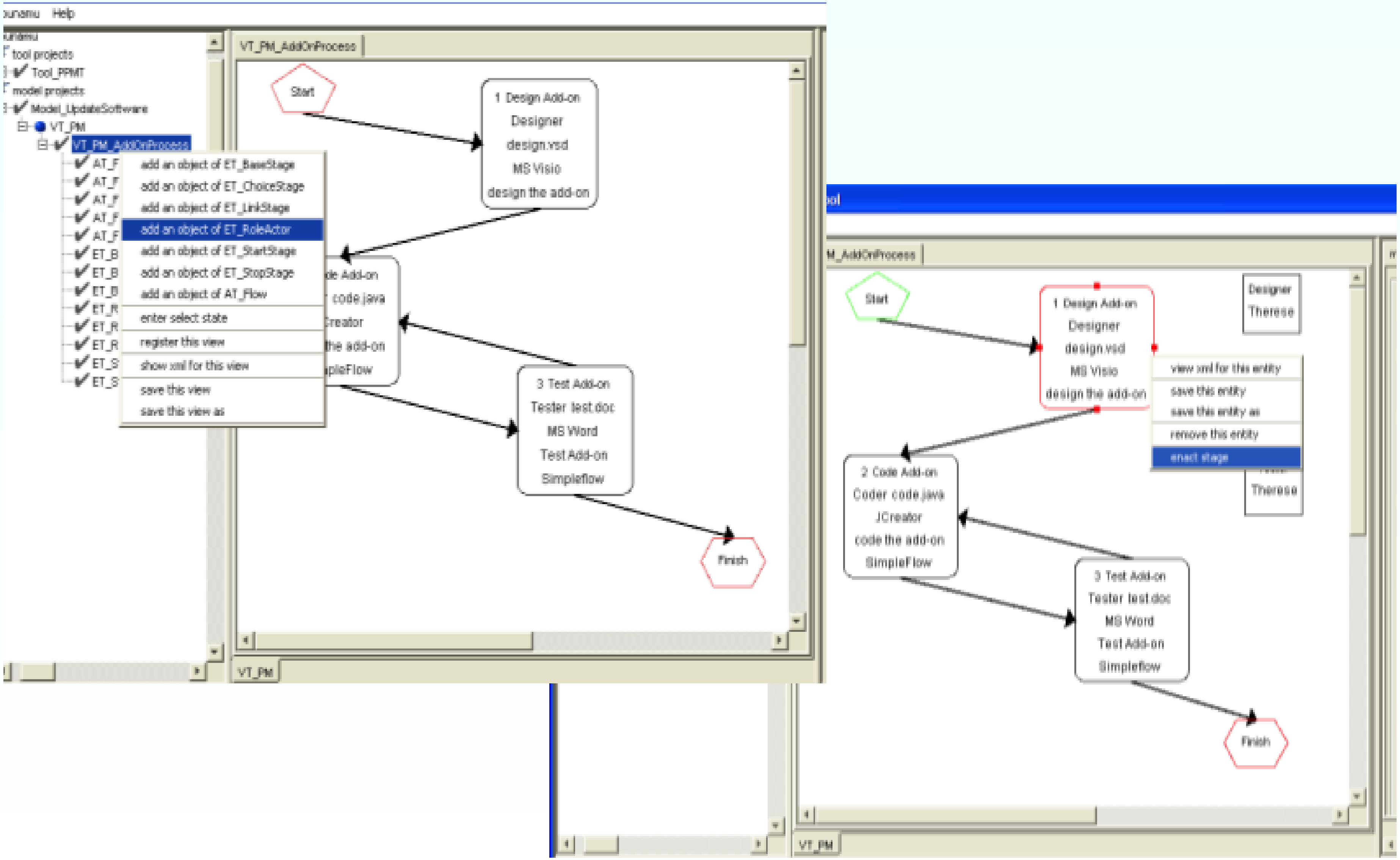
RoleActor

Choice

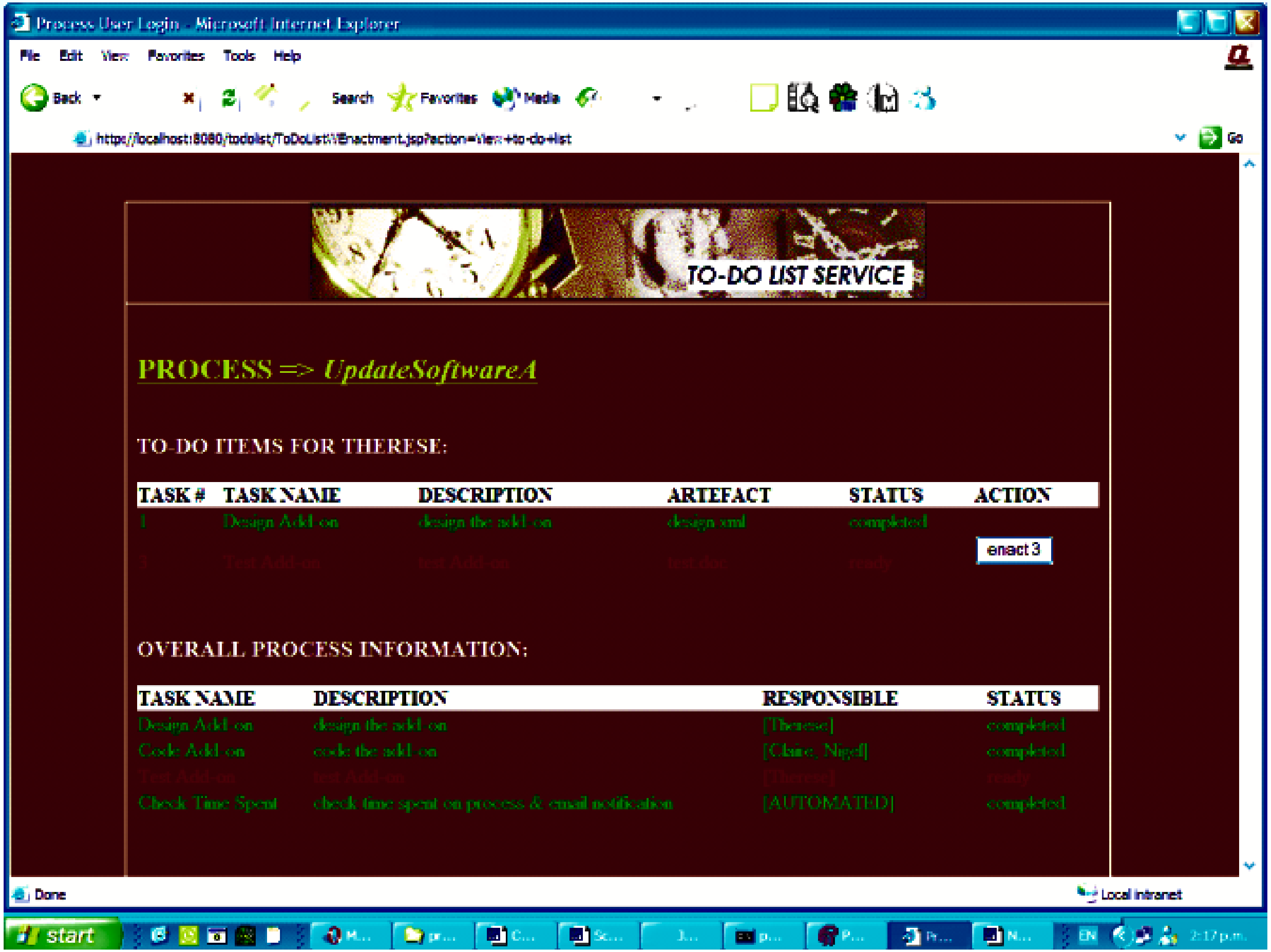
Resource

Stop stage

Process modelling & enactment



Web based to-do list



Process User Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
http://localhost:8080/todolist/ToDoList/Enactment.jsp?action=view+to-do-list
Go
TO-DO LIST SERVICE
Done
Local intranet
start

PROCESS => *UpdateSoftwareA*

TO-DO ITEMS FOR THERESE:

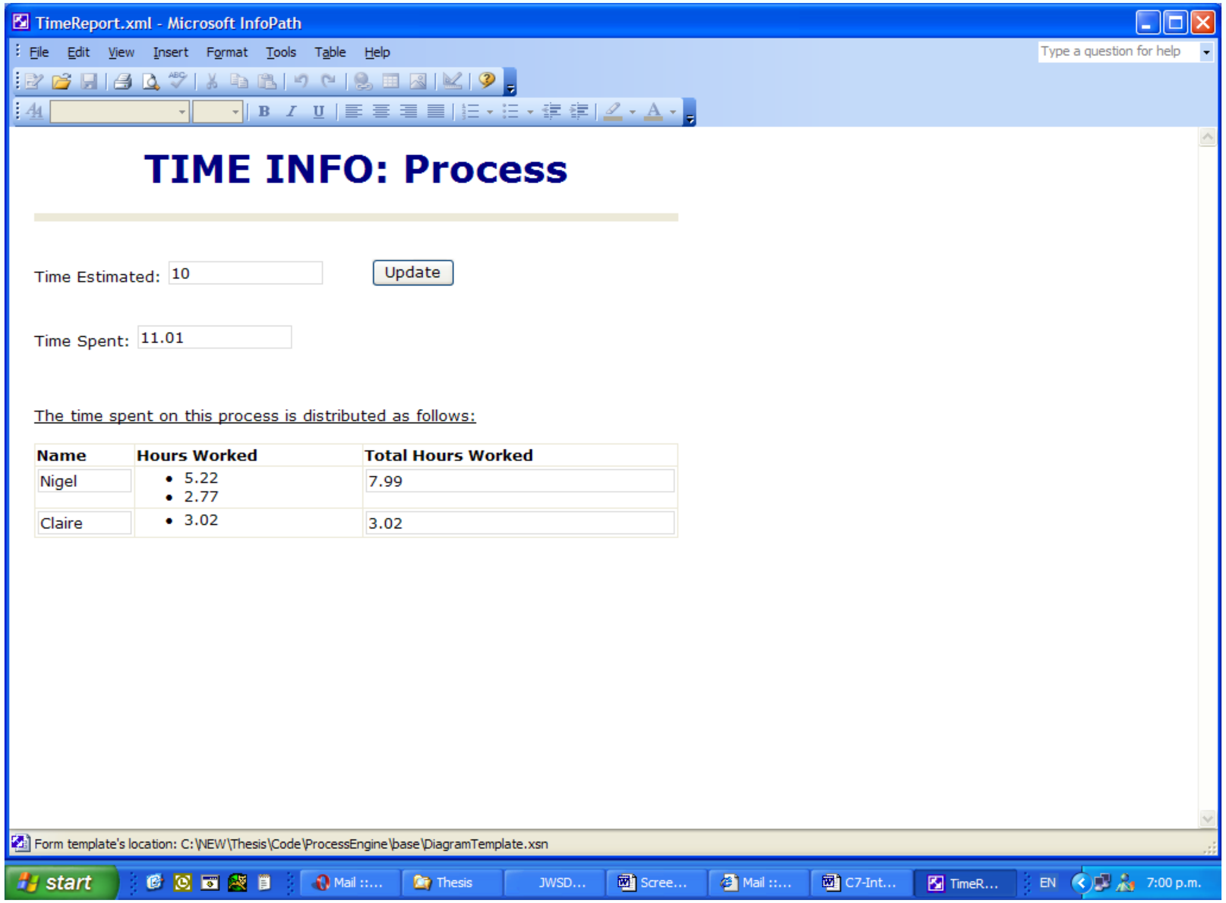
TASK #	TASK NAME	DESCRIPTION	ARTEFACT	STATUS	ACTION
1	Design Add-on	design the add-on	design.xml	completed	
3	Test Add-on	test Add-on	test.doc	ready	enact 3

OVERALL PROCESS INFORMATION:

TASK NAME	DESCRIPTION	RESPONSIBLE	STATUS
Design Add-on	design the add-on	[Therese]	completed
Code Add-on	code the add-on	[Claire, Nigel]	completed
Test Add-on	test Add-on	[Therese]	ready
Check Time Spent	check time spent on process & email notification	[AUTOMATED]	completed

Document viewing and editing

- **Uses Microsoft InfoPath for form based interaction**
- **InfoPath server provides web based data i/o**



TIME INFO: Process

Time Estimated: 10 Update

Time Spent: 11.01

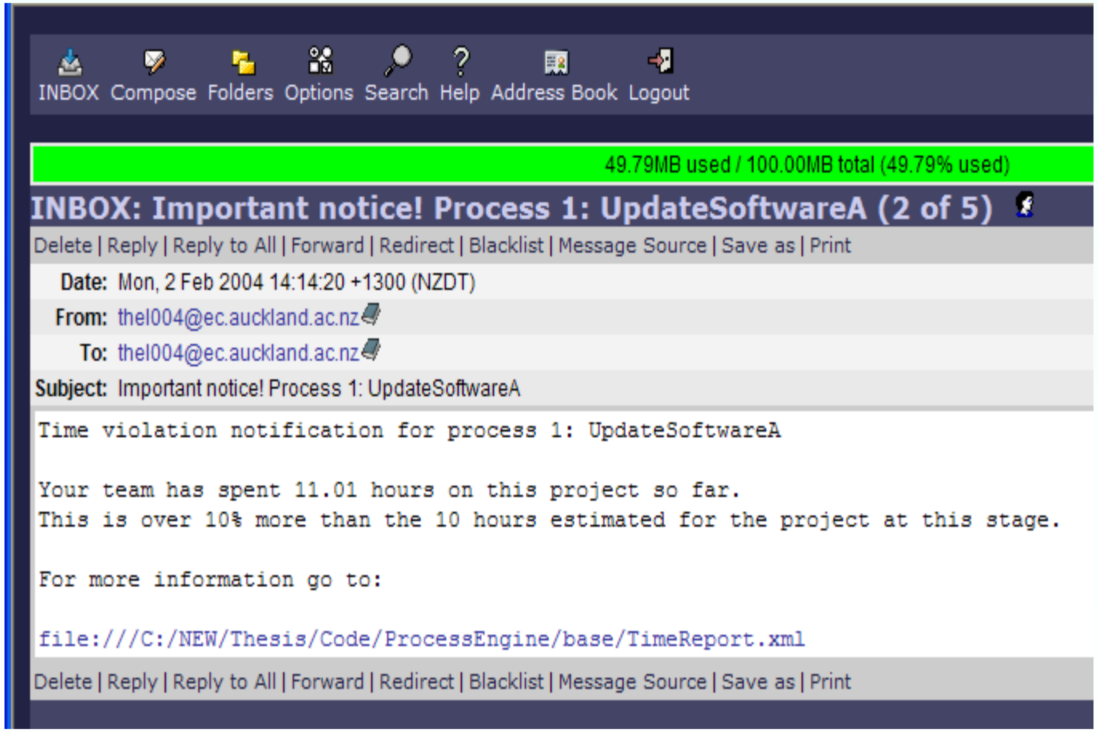
The time spent on this process is distributed as follows:

Name	Hours Worked	Total Hours Worked
Nigel	• 5.22 • 2.77	7.99
Claire	• 3.02	3.02

Form template's location: C:\NEW\Thesis\Code\ProcessEngine\base\DiagramTemplate.xsn

Rule based decisions

- **Uses Idiom as an alternative to simple flow engine**
- **Provides complex rule-based processing logic**
- **Can also automate activities based on rule-based decision making**
  - Eg send an email if a computed trigger value is above a threshold


INBOX
Compose
Folders
Options
Search
Help
Address Book
Logout
49.79MB used / 100.00MB total (49.79% used)
Delete | Reply | Reply to All | Forward | Redirect | Blacklist | Message Source | Save as | Print

INBOX: Important notice! Process 1: UpdateSoftwareA (2 of 5)

Date: Mon, 2 Feb 2004 14:14:20 +1300 (NZDT)

From: thel004@ec.auckland.ac.nz

To: thel004@ec.auckland.ac.nz

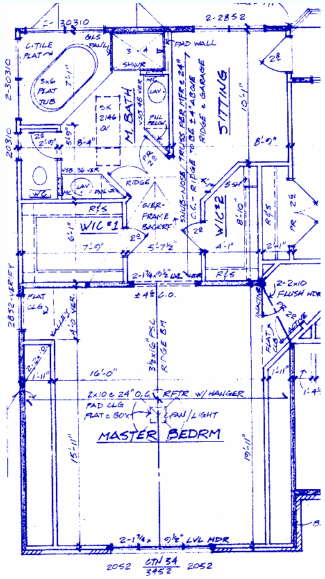
Subject: Important notice! Process 1: UpdateSoftwareA

```
Time violation notification for process 1: UpdateSoftwareA

Your team has spent 11.01 hours on this project so far.
This is over 10% more than the 10 hours estimated for the project at this stage.

For more information go to:

file:///C:/NEW/Thesis/Code/ProcessEngine/base/TimeReport.xml
```

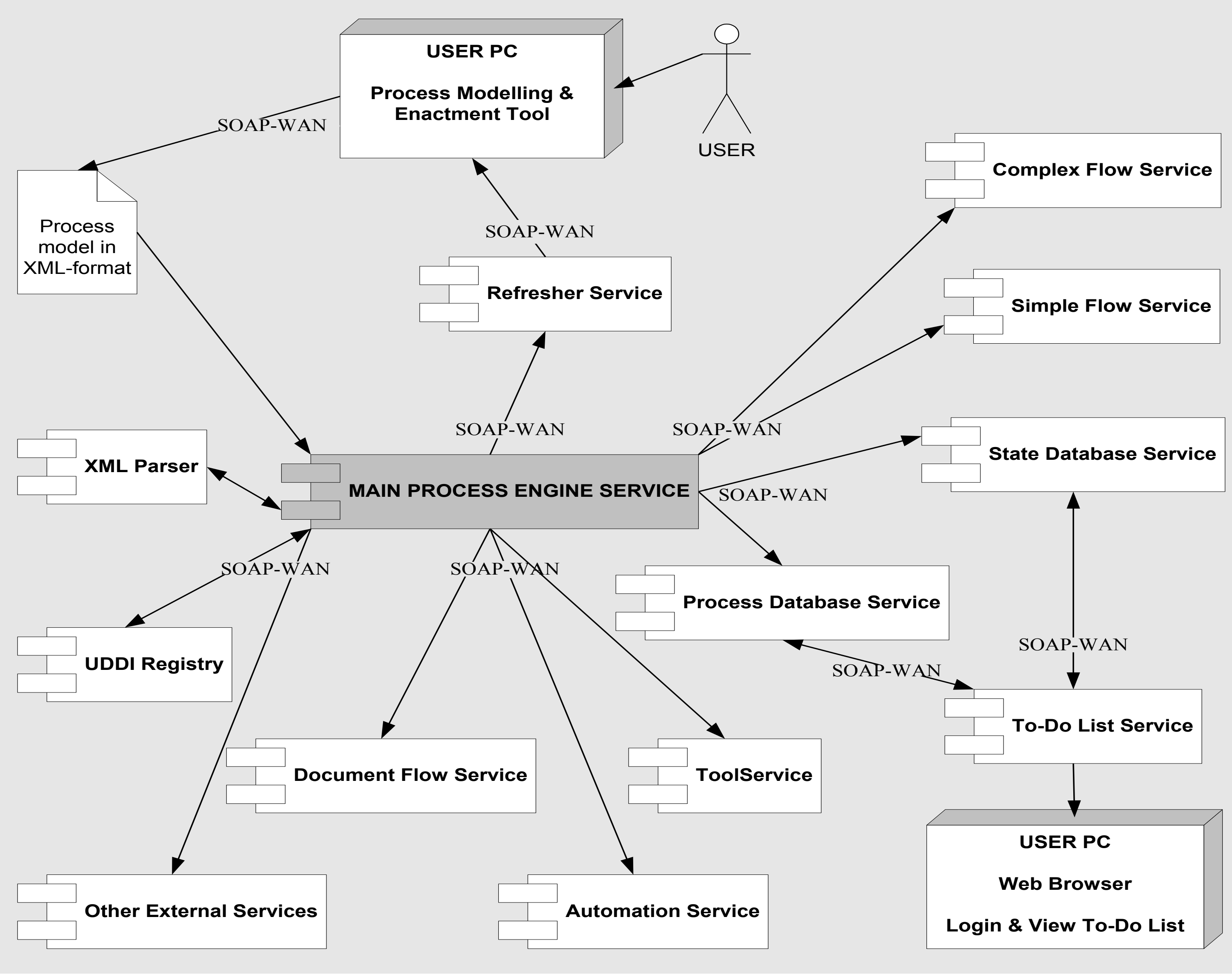
Delete | Reply | Reply to All | Forward | Redirect | Blacklist | Message Source | Save as | Print

Design and Implementation

- **Experimented with co-operating services to realise complex processing modelling and enactment tool**
- **Key components:**
  - Process modelling & enactment tool
  - Main process engine
  - Process & State Database services
  - Simple & complex flow services
  - Refresher & To do list services
  - Tool, automation & external services
  - Document flow service
  - UDDI registry



IMÅL’s Architectural Design



Implementation

- **Pounamu meta tool used to implement PPML**
  - Look and feel readily modifiable
  - Functionality extendible
- **Inter-component communications uses SOAP**
  - Java & JWSDP
- **SQL Server storage**
- **JSPs for web/thin client interfaces**

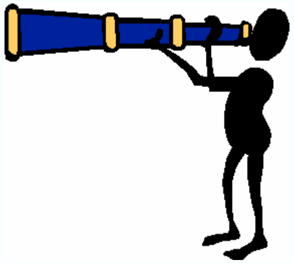
Evaluation

- **Three evaluations:**
- **User evaluation**
  - Favourable response, partic on usability
  - Some issues around simple/complex flow
  - Safety of look and feel mods
- **Cognitive dimensions**
  - See written paper
- **Performance test**
  - Can support over 100 concurrent users @ 1 enactment event/user/second
  - But dependent on automated task level

Conclusions & future work

- **Demonstrated feasibility of distributed service oriented process modelling and enactment approach**
  - Proof of concept implementation
  - Usability and performance shows good results

- **Future work**
  - Replicated instances of each service for fault tolerance
  - More versions of engine, notification services etc



References

- **Helland, T., Grundy, J.C. and Hosking, J.G. A Service-Oriented Architecture for Software Process Technology, In Proceedings of the 2006 Australian Conference on Software Engineering, Sydney, April 2006, IEEE CS Press.**
- **Yap, N., Chiong, H.C., Grundy, J.C. and Berrigan, R. Supporting dynamic software tool integration via web service-based components, In Proceedings of the 2005 Australian Software Engineering Conference, Brisbane, Australia, 29 March- 1 April 2005, IEEE CS Press.**
- **Zhu, N., Grundy, J.C., Hosking, J.G., Liu, N., Cao, S. and Mehra, A. Pounamu: a meta-tool for exploratory domain-specific visual language tool development, Journal of Systems and Software, Elsevier, vol. 80, no. 8, pp 1390-1407.**
- **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. Visual specification and monitoring of software agents in decentralised process-centred environments, International Journal on Software Engineering and Knowledge Engineering, Vol. 9, No. 4., August 1999, World Scientific Publishing Company, pp. 425-444.**
- **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., 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. 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.**