

# Supporting dynamic software tool integration via web service-based components

Nelson Yap<sup>1</sup>, Hau Chean Chiong<sup>1</sup>, John Grundy<sup>1,2</sup> and  
Rebecca Berrigan<sup>2</sup>

<sup>1</sup>*Department of Electrical and Computer Engineering*

<sup>2</sup>*Department of Computer Science*

*University of Auckland, New Zealand*

# Outline

- Motivation: the tool integration/extension problem
- Our approach: web service-based “toollets”
- Example of usage
- Architecture
- Evaluation
- Conclusions and Future Research

# Software Tool Integration and Extension



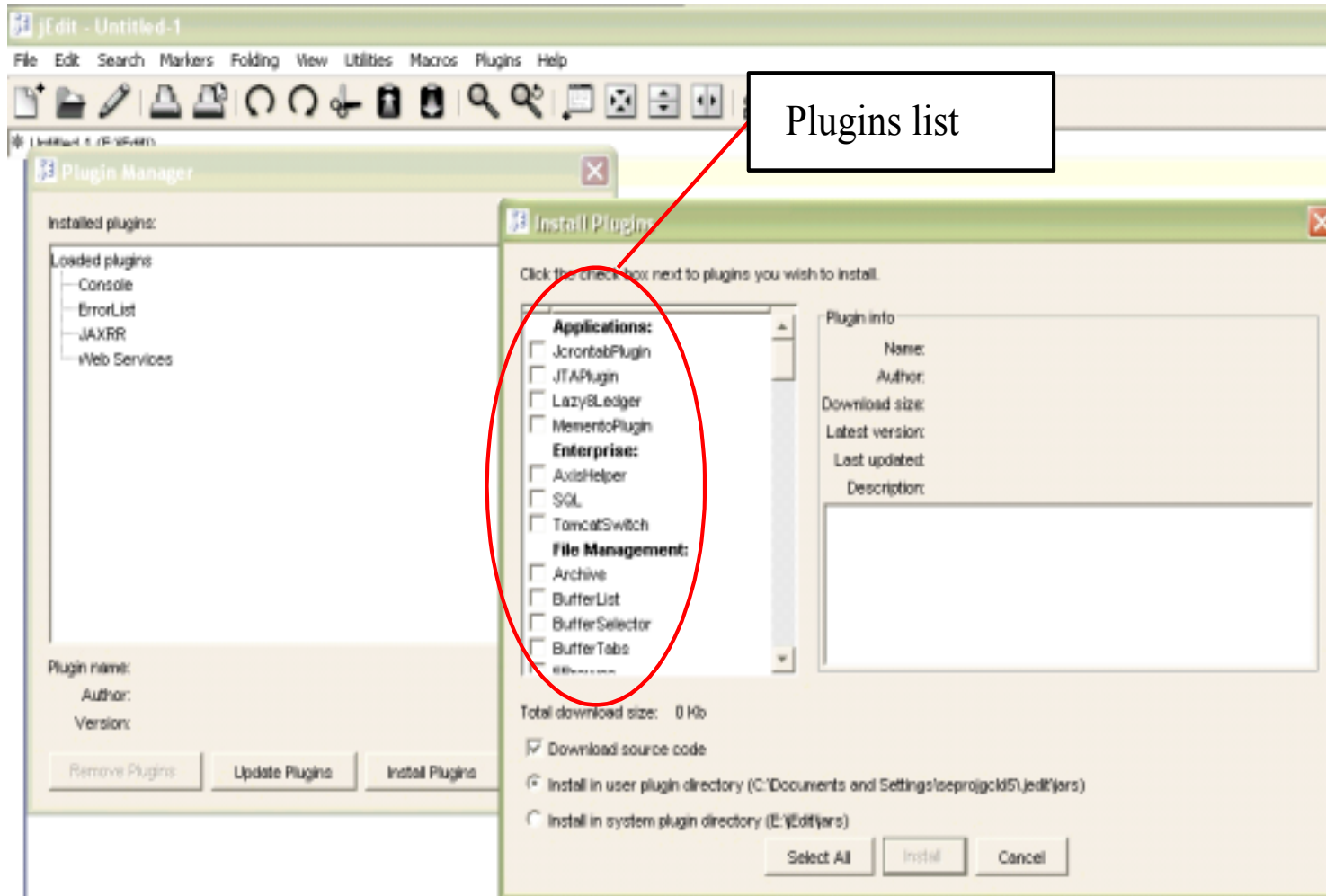
- Often a desire or even a need to integrate various software tools within an organisation
- Examples:
  - UML design tool -> code for Java IDE -> testing tool suite -> version control -> documentation tool
- Motivations:
  - Diversity of developer preferences - what aspects of an IDE to use vs more easily choosing alternatives
  - Opportunistic vs. Mandated Practice in organisation
  - Influence of software process on tool usage
  - Licensing issues for tools/tool extensions
  - Desire for “best of breed” facilities

2005  
YEAR

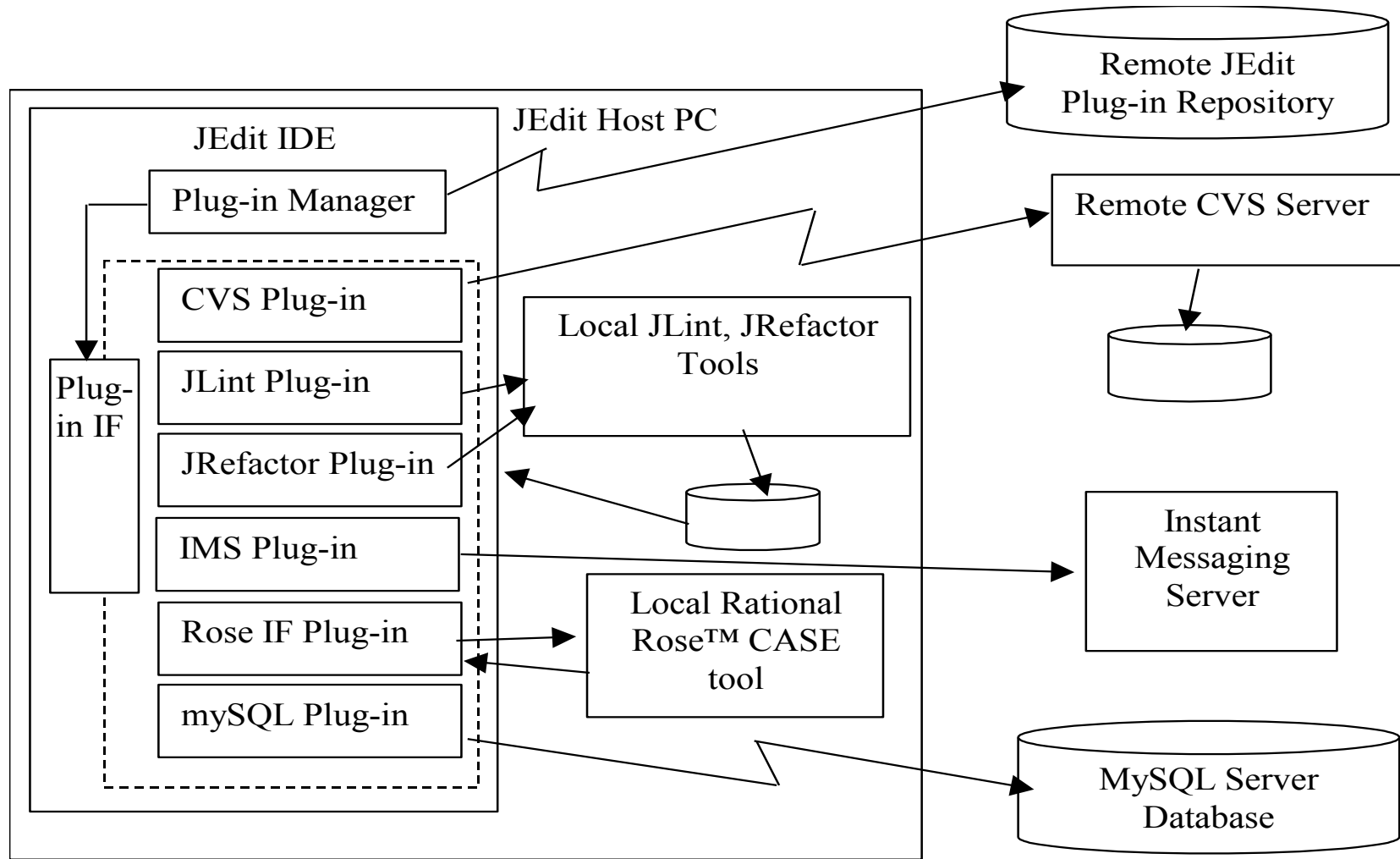
PRESENTATION

The University of Auckland | New Zealand

# Example: JEdit Plug-ins



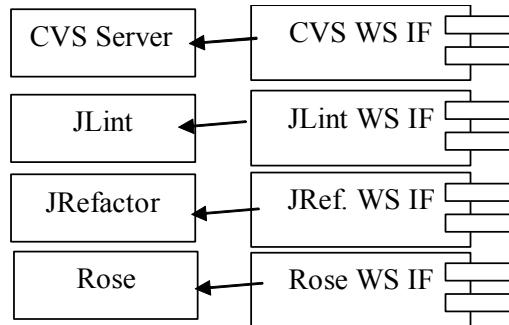
# Realisation: JEdit plug-in approach



# Current Approaches

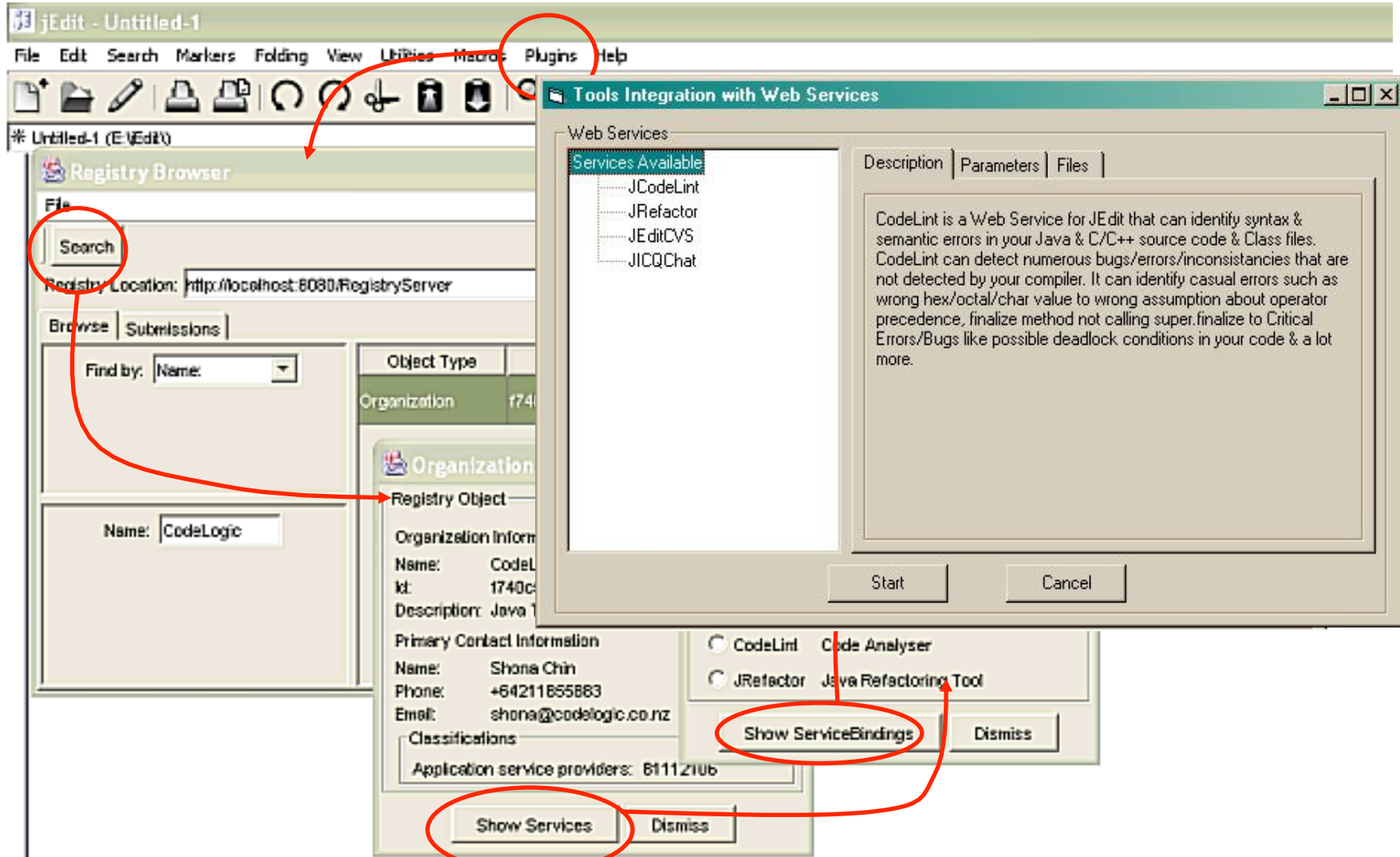
- Data integration (loosely coupled):
  - Shared file system/databases/version management tools
  - Exchange formats e.g. XMI
- Control integration (tightly coupled):
  - Typically API-based integration; plug-in managers
- User interface integration (loose to tight coupling):
  - Close integration of tool user interfaces e.g. HP SoftBench (wrapper) to plug-in approaches (shared)
- Process integration:
  - Various process-centred IDEs controlling when/how tools used

# Our Approach: JEdit-WS



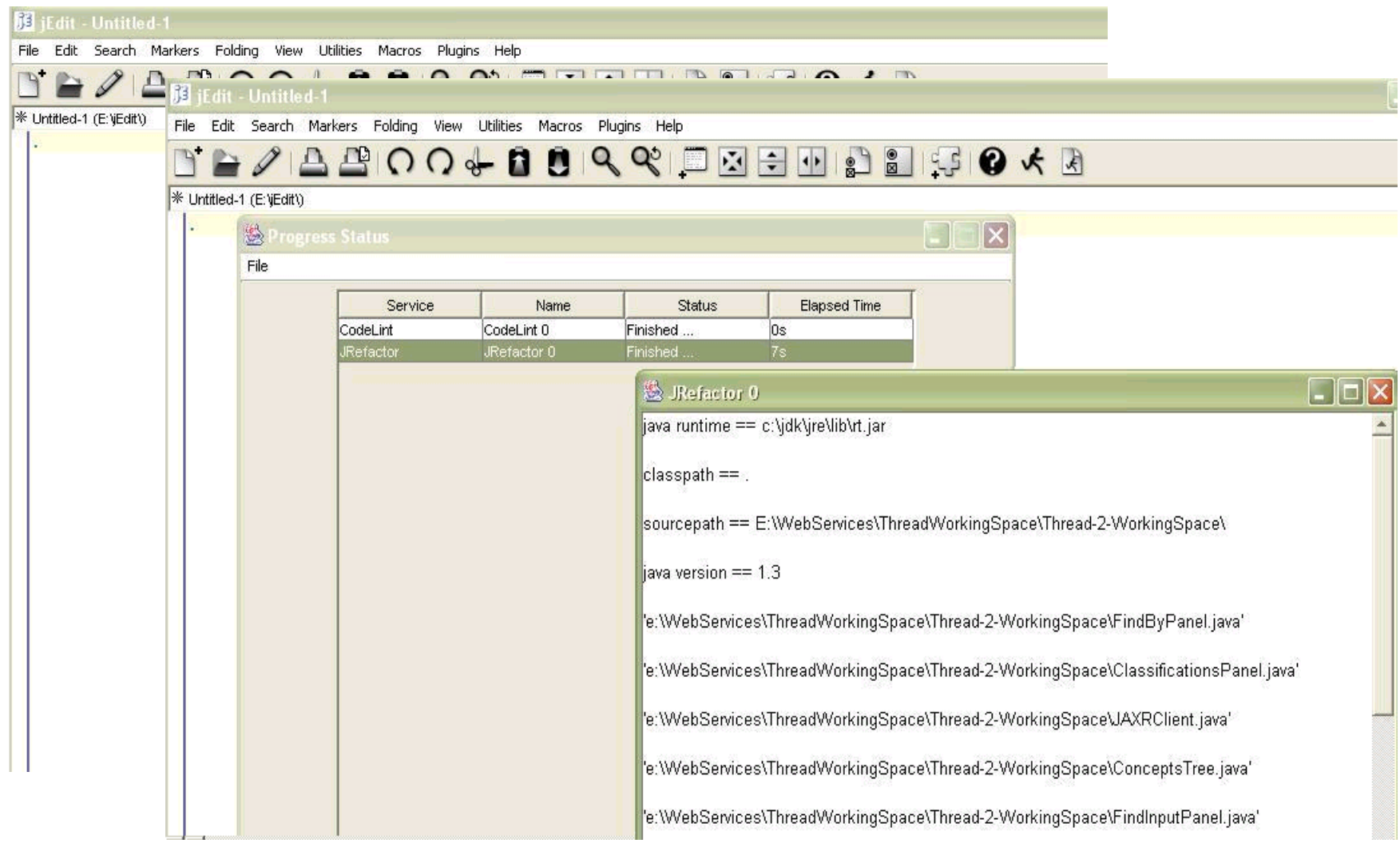
1. Develop remote WS  
comps for "tools"

# Example: Finding Toolets





# Refactoring



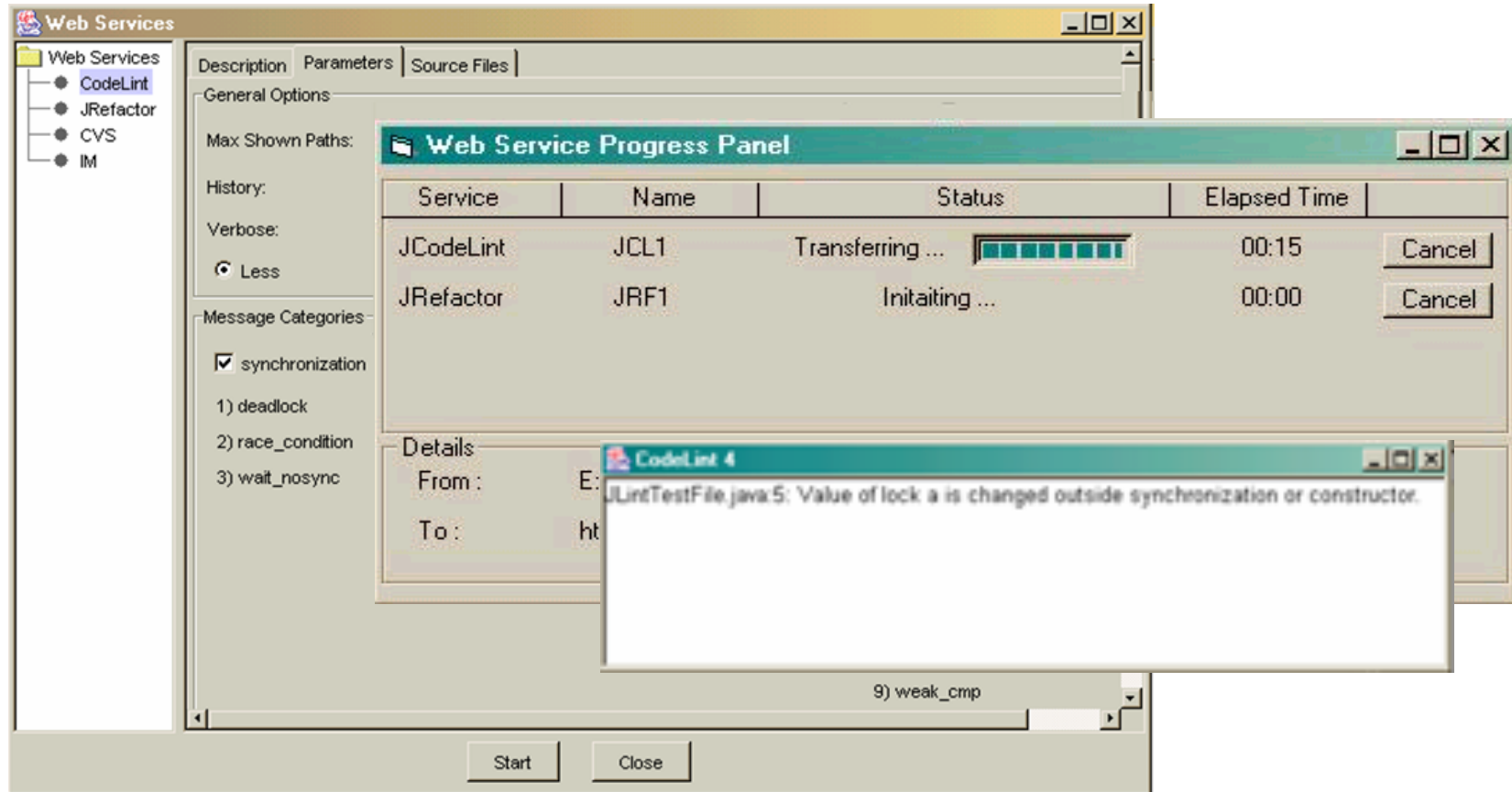
Progress Status

Service	Name	Status	Elapsed Time
CodeLint	CodeLint 0	Finished ...	0s
JRefactor	JRefactor 0	Finished ...	7s

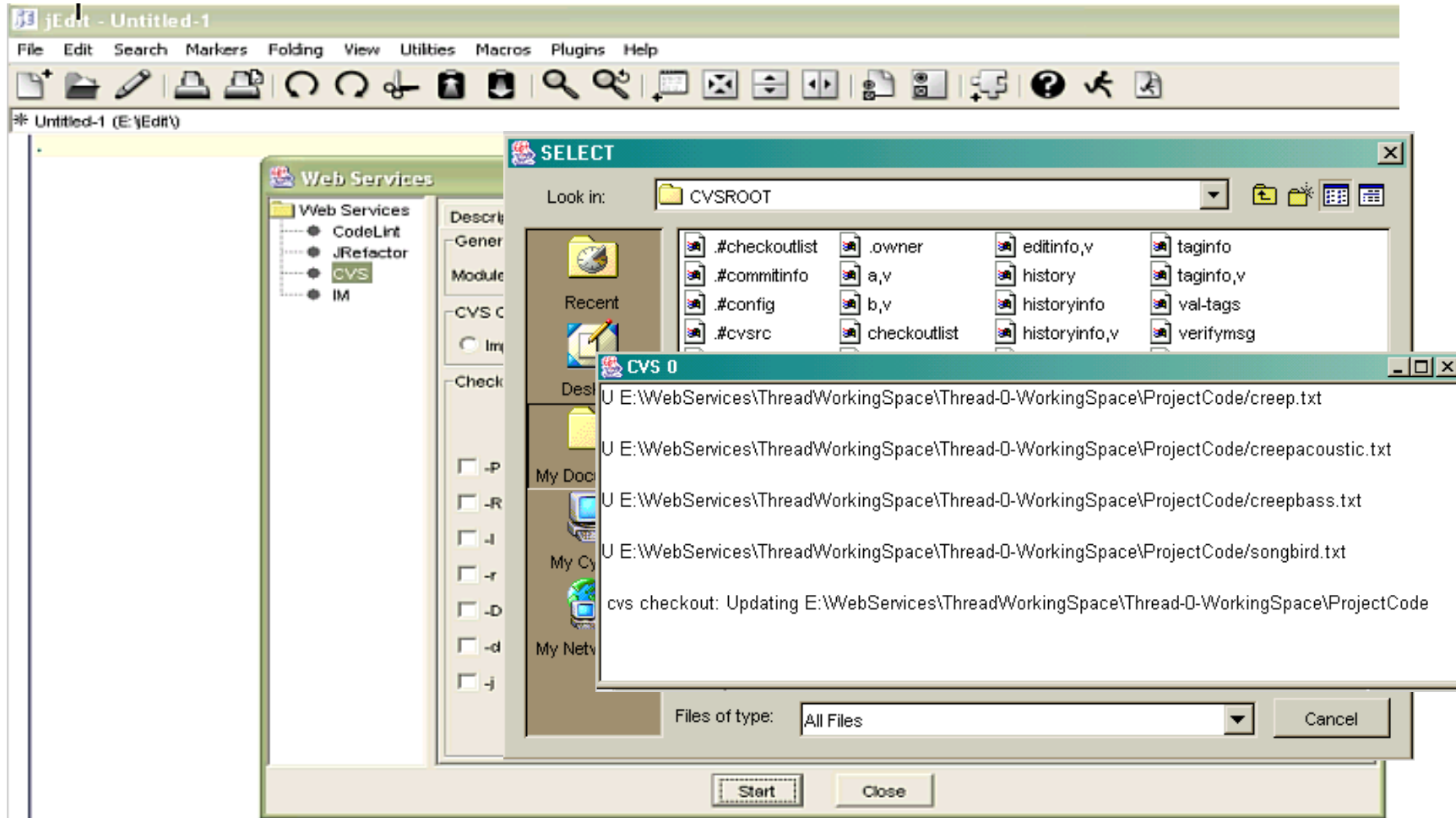
JRefactor 0

```
java runtime == c:\jdk\jre\lib\rt.jar
classpath == .
sourcepath == E:\WebServices\ThreadWorkingSpace\Thread-2-WorkingSpace\
java version == 1.3
'e:\WebServices\ThreadWorkingSpace\Thread-2-WorkingSpace\FindByPanel.java'
'e:\WebServices\ThreadWorkingSpace\Thread-2-WorkingSpace\ClassificationsPanel.java'
'e:\WebServices\ThreadWorkingSpace\Thread-2-WorkingSpace\JAXRClient.java'
'e:\WebServices\ThreadWorkingSpace\Thread-2-WorkingSpace\ConceptsTree.java'
'e:\WebServices\ThreadWorkingSpace\Thread-2-WorkingSpace\FindInputPanel.java'
```

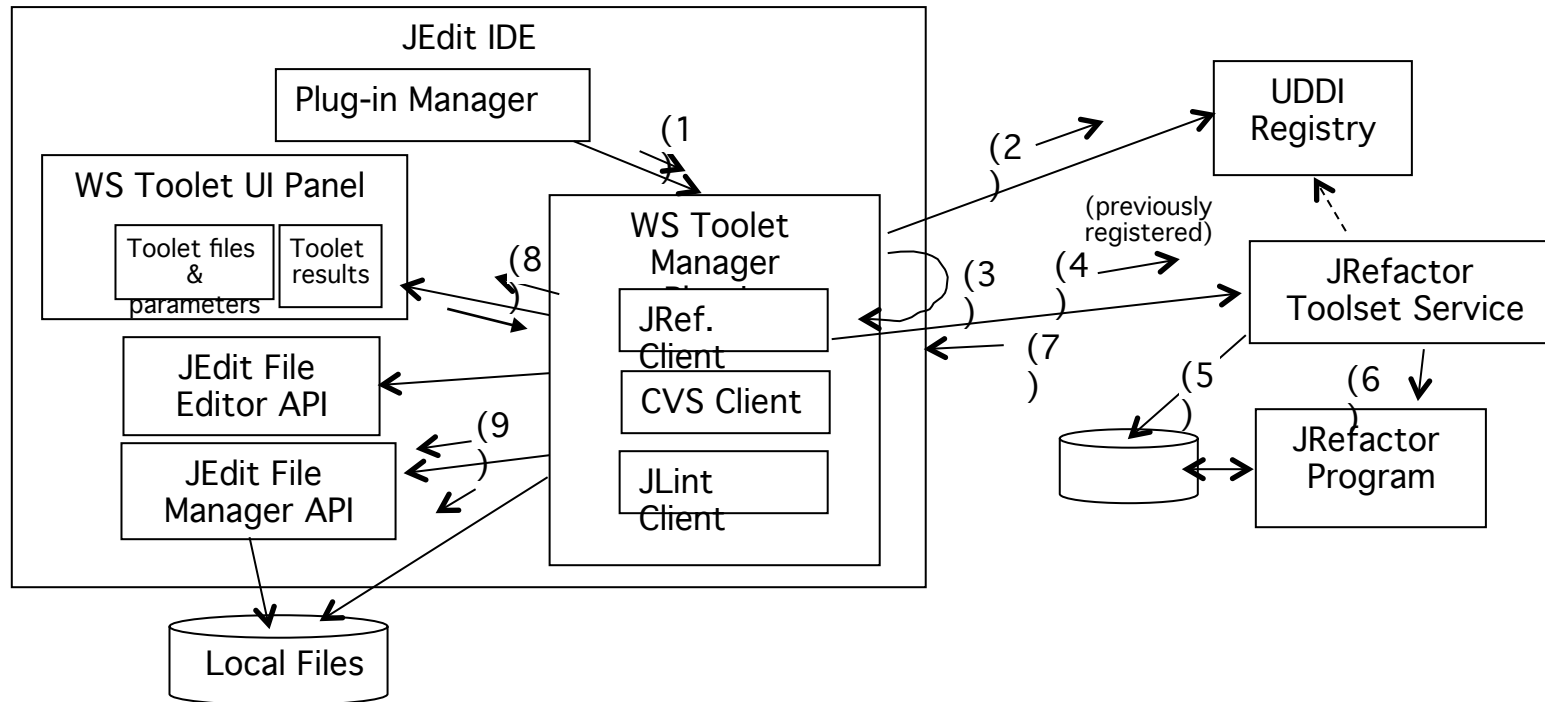
# Code Lint



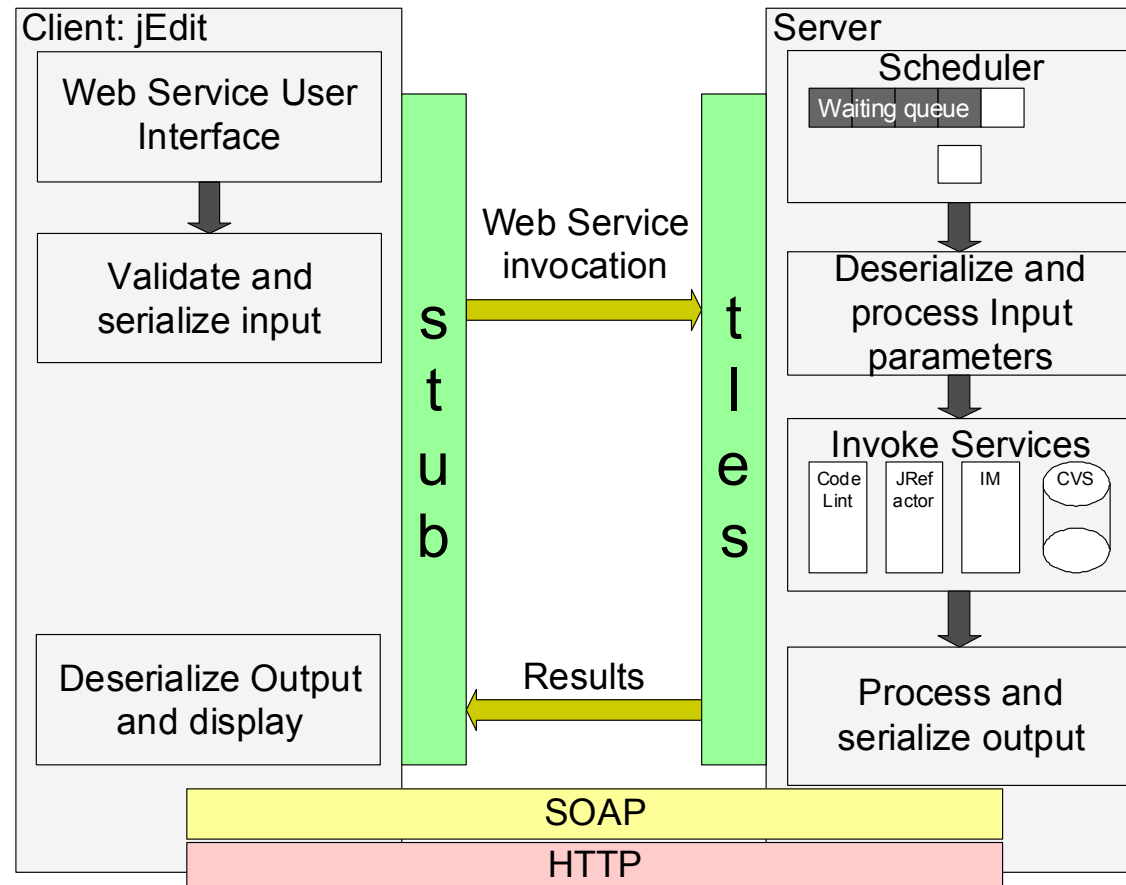
# CVS



# Operation of JEdit-WS



# Web service prototype



# Discovery Enhancement via “Aspects”

- Began extending the web service description language (WSDL) to better-characterise remote tool service characteristics
- Includes better support in tool description for:
  - User interface(s) to add to client tool
  - Transport mechanism, security protocols etc to use
  - Data management approaches - local, remote, caching policy etc
  - Other service resource utilisation issues
- Using aspect-oriented techniques to achieve - characterise service, discover service, integrate service

# Evaluation

- Group of experienced Java developers, familiar with the kinds of tool services available to JEdit-WS
- Tasks involved discovering appropriate services; integrating into JEdit-WS; using on programming task
- Feedback on discovery and integration very positive
- Some issues with user interfaces of integrated tools; some issues with remote tool invocation/reporting of results of processing
- Some concerns with performance for toolsets of certain classes
- Difficulty of building toolsets moderate (requires detailed knowledge of web services but not JEdit)

# Conclusions and Future Research

- Web services provide reasonable service-based infrastructure for extending IDEs in certain ways
- Suit toolet services that don't have high user response requirement
- Allows choice of toolet services, sharing of toolets, reduces some upgrade and consistency issues
- Need to better characterise services and integrate new user interfaces better into IDE client
- Data management, communication, security issues need further research work
- Need support for developing and testing toolets



# References

- 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.
- Grundy, J.C., Ding, G., and Hosking, J.G. Deployed Software Component Testing using Dynamic Validation Agents, Journal of Systems and Software: Special Issue on Automated Component-based Software Engineering, vol. 74, no. 1, January 2005, Elsevier, pp. 5-14.
- 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.
- 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., 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.