



Towards a Set of Application Independent Clustering Criteria within an Architecture Recovery Approach

Aline Vasconcelos

Claudia Werner

{aline,werner}@cos.ufrj.br

Reuse Project: <http://reuse.cos.ufrj.br/odyssey>



Motivations

- Legacy systems in general do not have an up-to-date architectural documentation that can support system maintenance, reengineering, reuse etc. Therefore **architecture recovery** is needed.

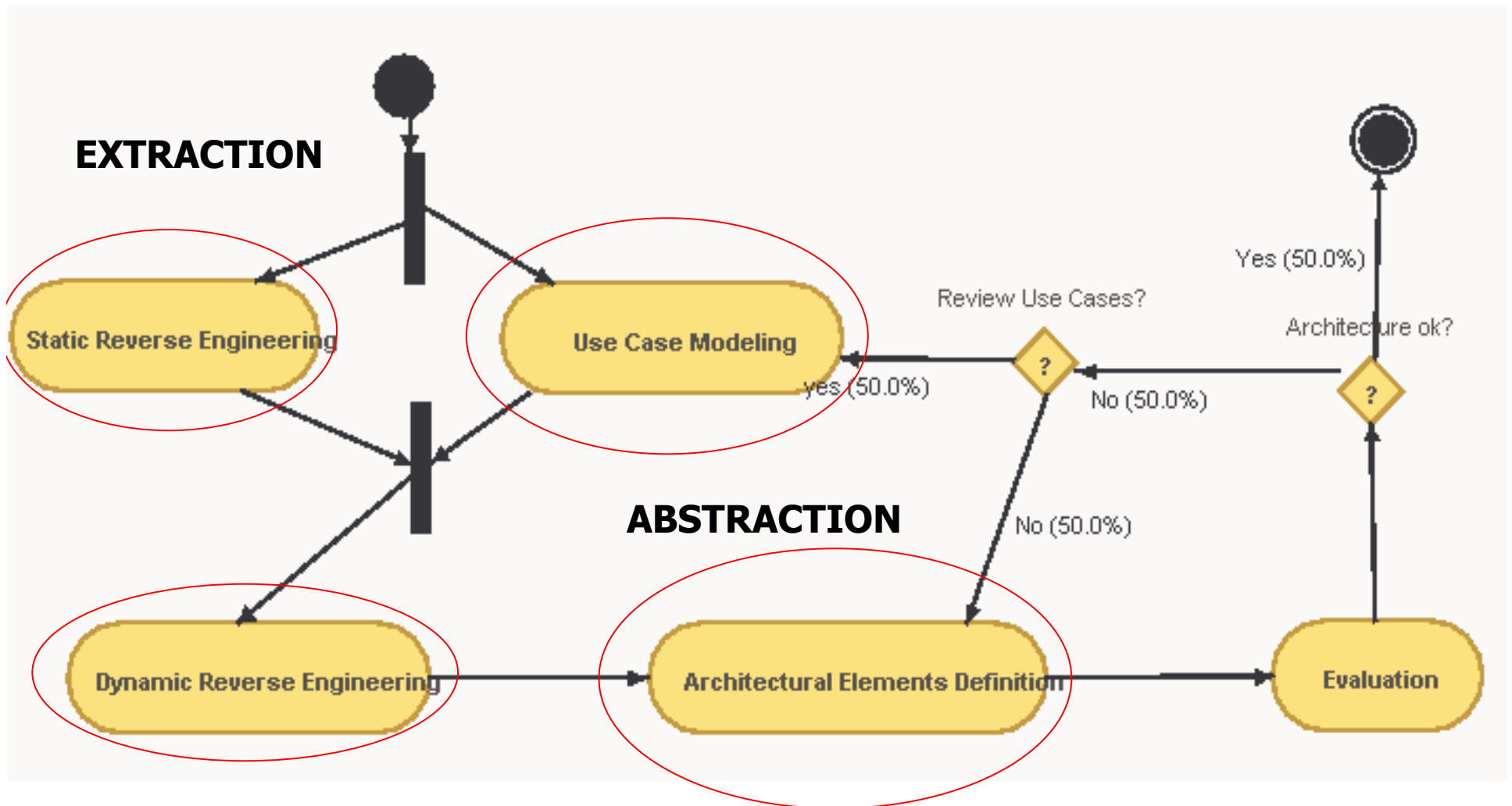


Proposal and Goals

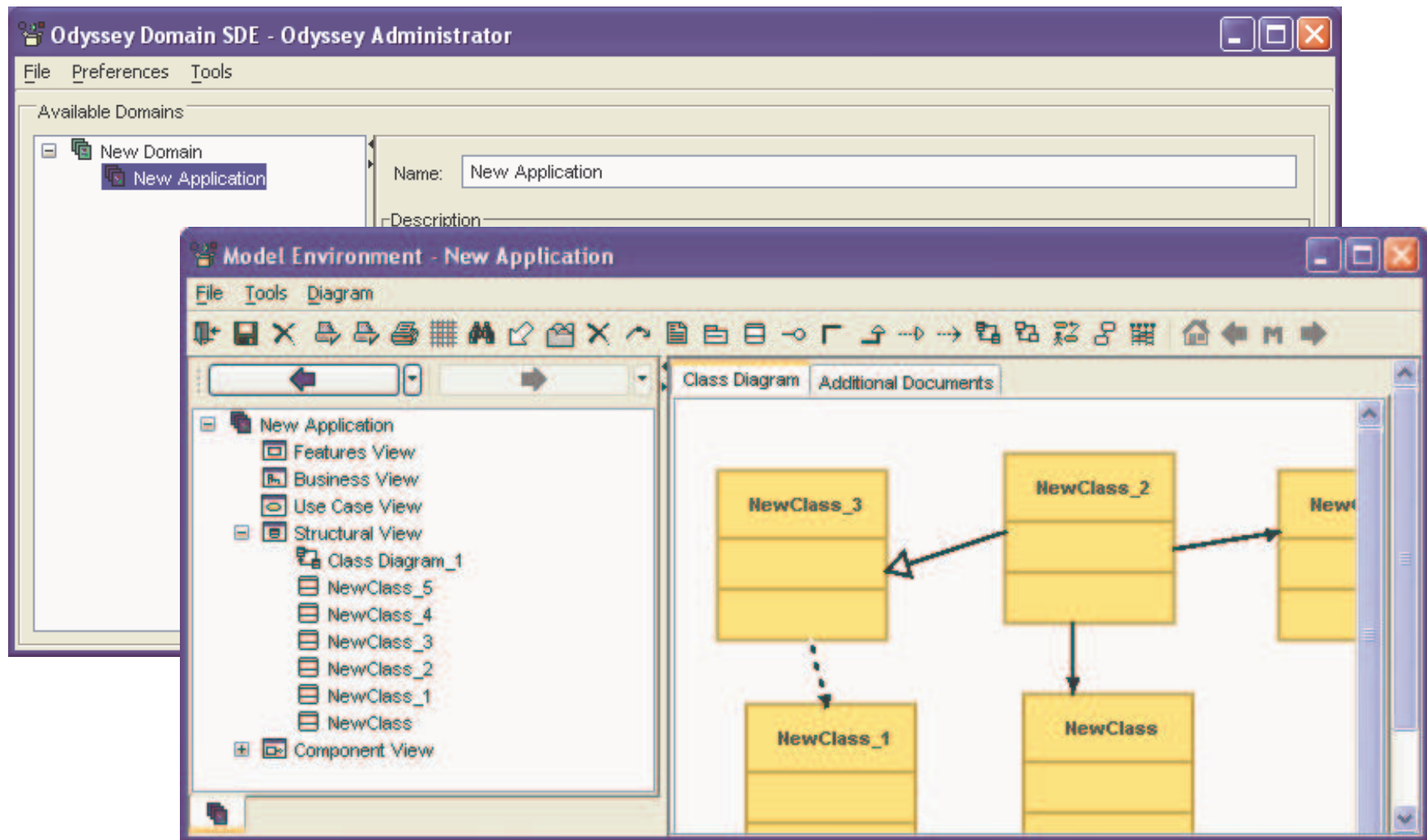
- Our goals:
 - To propose a set of **application independent clustering criteria** to architectural elements reconstruction, that can support architecture recovery.
 - Recovery of architectural elements based on the functionality or services they provide.

Proposed Process to Architecture Recovery

(Vasconcelos e Werner, 2004)



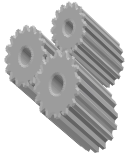
Context: Odyssey Environment - a reuse based software development environment.





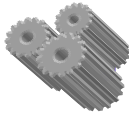
Static and Dynamic Reverse Engineering

- **Static Analysis** aims at the extraction of a **Class Model** from Java source code.



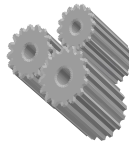
Ares Tool - plugin to the Odyssey environment.

- **Dynamic Reverse Engineering** – 2 Steps:
 - **Execution Traces Collection**, guided by use-case scenarios.



Tracer Tool (AspectJ).

- **Sequence Diagrams Extraction**



Phoenix Tool – a plugin to Odyssey.



Clustering Criteria to Architectural Elements Reconstruction

- The **criteria** are mainly based on **dynamic analysis**, since object interactions are crucial in object oriented systems.
- **Data mining** is used to detect classes that support common functionality.
- **Static analysis is also employed** in order to complement the results of the dynamic analysis.

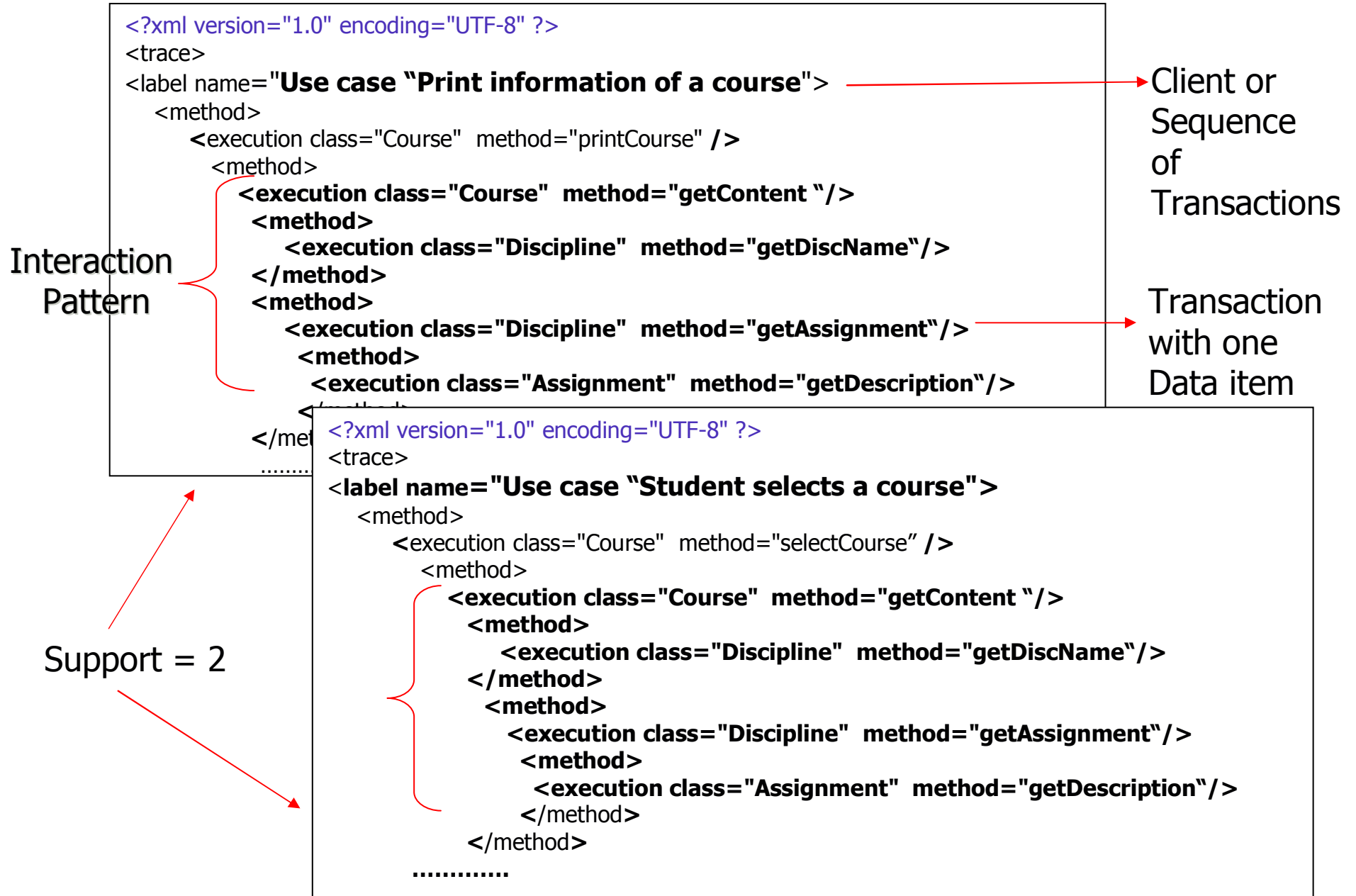


Clustering Criteria to Architectural Elements Reconstruction

- **Criterion 1:** Cluster **classes** that participate in **interaction patterns** with support greater or equal to a minimum support treshold value into an **architectural element**.
 - **Interaction Pattern** = sequence of methods frequently invoked during system execution.

Mining Sequential Patterns

AprioriAll (Agrawal e Srikant, 1995)





Mapping of Concepts from the Database Domain

- Use-case scenarios (i.e. execution traces) => sequences of client transactions in databases.
- Method executions => transactions containing only one item.
- Support for an interaction pattern => number of use-case scenarios in which it appears.

Goal: to detect classes that implement a common behavior in the architecture.

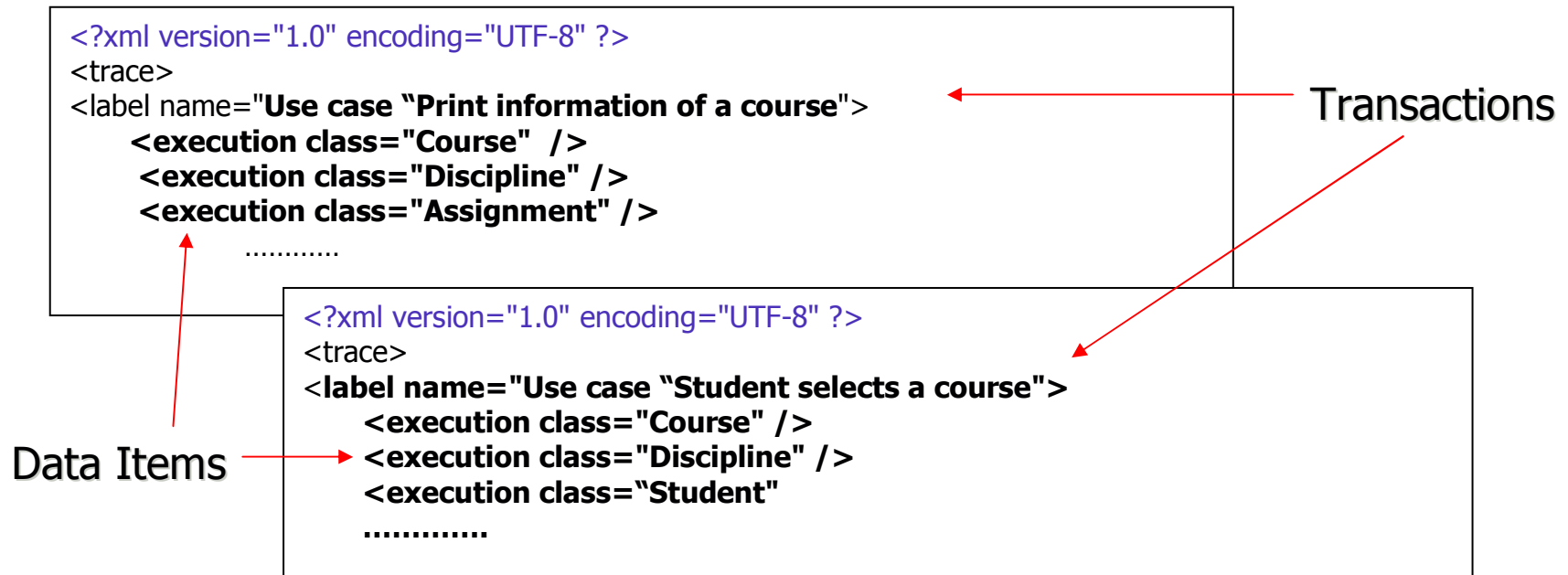


Clustering Criteria to Architectural Elements Reconstruction

- **Criterion 2:** Cluster **classes** that participate in **association rules** with support greater or equal to a minimum support treshold value and with confidence greater or equal to a minimum confidence value into an **architectural element**.
 - **Association Rule:** Class a => Class B

Mining Association Rules

Apriori (Agrawal e Srikant, 1994)



Association Rule: Course => Discipline

Support: 2

Confidence: 100%

Association Rule: Course => Assignment

Support: 1

Confidence: 50%



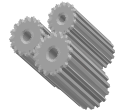
Mapping of Concepts from the Database Domain

- Use-case scenarios (i.e. execution traces) => database transactions.
- Class occurrences in the execution traces => data items.
- Support s => $s\%$ of the transactions in the database contains $A \cup B$.
- Confidence c => percentage of transactions containing one item A in which another item B appears. In this case we say that $A \Rightarrow B$ with confidence of $c\%$.
- Goal: to detect strongly related classes that implement common system functionalities.



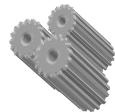
Clustering Criteria Based on Static Analysis

- **Criterion 3:** Cluster classes that present a common String in their names.



QueryClasses – plugin to Odyssey.

- **Criterion 4:** Cluster classes with a high coupling value among them.



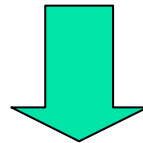
MetricTool – plugin to Odyssey.



Visualization – “4+1 View” Model

(Kruchten, 1995)

- **Logical View:** UML Package Diagram
- **Process View:** Sequence Diagrams at the Architectural Level
- **Scenarios View:** Sequence Diagrams associated to Use-Case Scenarios



Odyssey Environment



Conclusions and Future Work

- Contributions:

- Independent clustering criteria to architectural elements reconstruction, being reusable in different contexts.
- Recovery of architectural elements based on their functionality or services.
- Criteria support the decision making in architecture recovery.

- Ongoing and Future Work:

- The approach has been applied in two case studies involving software tools. We are planning its application in an industrial setting.
- Refine our process and criteria.



References

- GANNOD, G., 1999, "Automated Reverse Engineering and Design Recovery", *14th IEEE International Conference on Automated Software Engineering*, Florida, USA, October, Tutorial 3.
- A.V. Deursen., C. Hofmeister, R., Koschke, L. Moonen, and C. Riva (2004) "Symphony: View-Driven Software Architecture Reconstruction", Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), Oslo, Norway, June, p. 122-132.
- A. P. V., Vasconcelos, C.M.L., Werner (2004) "Software Architecture Recovery based on Dynamic Analysis", 18th Brazilian Symposium on Software Engineering, 1st Workshop on Modern Software Maintenance, Brasilia, DF, Brazil, October.
- *R. Agrawal, and R. Srikant (1994) "Fast Algorithms for Mining Association Rules", Proc. 20th Very Large Databases Conference (VLDB), Santiago, Chile, September, p. 487-499.*
- R. Agrawal, and R. Srikant (1995) "Mining Sequential Patterns", *Proc.11th International Conference on Data Engineering (ICDE'95)*, Taipei, Taiwan, March, p. 3-14.



References

- P.B. Kruchten (1995) "The 4+1 View Model of Architecture", *IEEE Software*, Vol. 12, Number 6, November, p. 42-50.
- D.R. Harris, H.B. Reubenstein, A.S. Yeh (1995) "Reverse Engineering to the Architectural Level", *Proc. 17th International Conference on Software Engineering*, Seattle, Washington, April, p. 186-195.
- R. Kazman, and S.J. Carrière (1997) "Playing Detective: Reconstructing Software Architecture from Available Evidence", Technical Report CMU/SEI-97-TR-010, *Carnegie Mellon University*, October.
- K. Sartipi, and K. Kontogiannis (2003) "On Modeling Software Architecture Recovery as Graph Matching", *Proc. 19th International Conference on Software Maintenance*, Amsterdam, Netherlands, September, p. 224-234.
- L. Ding, and N. Medvidovic (2001) "Focus: a LightWeight, Incremental Approach to Software Architecture Recovery and Evolution", *Working IEEE/IFIP Conference on Software Architecture*, Amsterdam, Netherlands, August, p. 191-200.
- C. Riva, and J.V. Rodriguez (2002) "Combining Static and Dynamic Views for Architecture Reconstruction", *Sixth European Conference on Software Maintenance and Reengineering (CSMR'02)*, Budapest, Hungary, March, p. 47-56.



Towards a Set of Application Independent Clustering Criteria within an Architecture Recovery Approach

Aline Vasconcelos

Claudia Werner

{aline,werner}@cos.ufrj.br

Reuse Project: <http://reuse.cos.ufrj.br/odyssey>