Software Architecture Recovery Using Data Mining Techniques

**Approach**

We propose a framework for software architecture recovery and restructuring in this framework, the user specifies a high-level abstraction of the system using a structured pattern language (see  [Architecture Query Language (AQL)](#)).

Then, a pattern matching engine provides an optimal match between the given pattern and a decomposition of the legacy system entities into modules while satisfying the inter/intra-module constraints defined by the pattern.

We propose a framework for software architecture recovery and restructuring.

*Major approaches:*

1. Should relate to specific re-engineering requirements
2. Need to handle the high level architectural styles
3. Should be scalable to handle large systems

*Pre-process*

- Extraction of high-level information from the source code
- Conversion of high-level information to a source model

*Post-process*

- Application of the source model to the target system
- Recovery of the source model

**Data Mining Technique (Apriori)**

- Discovery of interesting and non-trivial relations among entities in the data
- Apriori is a fundamental data mining algorithm (Apriori)

**Application of Data Mining in Architecture Recovery**

1. **Discovery of Interesting and Non-trivial Relations among Entities**
   - *Apriori* algorithm is used to identify interesting and non-trivial relations among entities.

2. **Example of Apriori Algorithm**
   - Given a database of entities, the Apriori algorithm is used to find frequent itemsets.

**Software Architecture Recovery**

- **Definition:** Extracting high-level information from some low-level software representation such as source code.
- **Categories:**
  1. Constitutes a major task in software maintenance
  2. Should relate to specific re-engineering requirements
  3. Need to handle the high level architectural styles
  4. Constraint satisfaction to satisfy user-defined constraints
  5. Query-based techniques based on specialized queries and high-level architectural styles

**Case Study: CLIPS**

- **40 KLOC in C**
- **User Interfaces:**
  - Web browser (Netscape)
  - Hypertext links to actual entities in the source files
  - Various information: distribution of recovered entities into files, browsing the query, statistical data for link-constraint violation, links between modules

**Model of the Matching Process**

- The matching process selects the entities for each module based on high association value and high average clustering value to the group of entities already selected for the module.

**Assigning Entities to Modules**

- Each module has one (or more) main-seeds which determine the domain of entities to be put in the module, and zero or more seeds which specialize the query.

**Solution**

- The graph of a recovered module with 18 functions using RIGI graph visualizer:
  - The graph of a recovered module with 10 functions using RIGI graph visualizer
  - A query and its solution represented using Netscape browser

**Contact:**

Kamran Sartipi
Kostas Kontogiannis
University of Waterloo
{ksartipi, kostas}@swen.uwaterloo.ca

**References**

- [Architecture Query Language (AQL)](#)
- [Web browser (Netscape)](#)