Scenario for Architectural Recovery Based on Pattern Matching

A branch and bound search algorithm with association-based score function is used. The method is known as inexact graph matching. The architectural recovery process tries to find a series of graph edit operations on an input graph to match a given pattern graph. This results in incremental reconstruction of the system architecture.

Definition: Extracting high-level information from some low-level, software representation such as an executable.

Graph Matching Process
- At each phase of the matching process, G', incremental result graph G' is incrementally matched against G. Incremental result graph G' results in incremental reconstructed.
- We perform graph edit operations on G' to modify G' until it matches the given pattern graph. This results in incremental reconstructed.

Example of attributes in software system:
- Label: A unique string for nodes only.
- Type: Identifiers to classify nodes and edges.
- Location: Two images for self and link edge.

Software System as Attributed Relational Graph
An ARG is a six-tuple G = (N, R, A, E, f, g):
- N = (n, n, ... n): attributed vertices (nodes)
- R = (r, r, ... r): directed attributed edges (relations)
- A & E: alphabets for node & edge attributes
- f: node & edge labeling functions

Example of attributes in software system:
- Label: A unique string for nodes only.
- Type: Identifiers to classify nodes and edges.
- Location: Two images for self and link edge.

Architecture Query Language (AQL)

```
import VARIABLES:
 V1, V2, V3, V4, V5, V6, V7, V8, V9

import TYPES:
 T1, T2, T3, T4

import FUNCTIONS:
 F1, F2, F3, F4, F5

import AS: attributes

VARIABLES:        var     $CV(3 .. 5)
TYPES:                type   $CT(0 .. 2)
FUNCTIONS:      func   ?EF
LOCATION:        loc     1, 2, 3, 4
```

Graph Matching Example

```
Input graph

Pattern graph

Graph Matching
```

Experiment with Xfig 75 KLOC:

```
Pattern ... Query ... Architecture
```

```
Result Xfig: Netscape browser and Rigi graph visualizer
```

A Query is modeled as a multi-graph of nodes and edges:
- Each node represents an abstract module to be instantiated with system entities.
- Each edge represents a group of link constraints between two modules in the form of import/export of resources (func / type/var).
- 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.

Graph Matching
- F: G ? G' maps the nodes and relations of graph G onto graph G'.
- Exact graph matching:
- Inexact graph matching:
  - An optimal sequence of graph edit operations, each as: insertion / deletion / modification of nodes and edges of G so that G and G' become isomorphic.