Modeling XP Maintenance Activities Using Random Graph

M.Marchesi, Università di Cagliari, Cagliari, Italy michele@diee.unica.it
G.Succi, University of Alberta, Edmonton, Canada Giancarlo.Succi@ee.ualberta.ca
N. Serra, Unversità di Cagliari, Cagliari, Italy philcore@tiscalinet.it

Introduction

Maintenance is a hard task. When a maintainer is asked for make a change to a large system, he has to understand how the changes will affect the whole system. Thus, it is important to identify that modules and entities within the system that are likely to be involved in the maintenance process. XP Maintenance should be fast and simple. We propose a Random Graph based model for representing system and a representation of the maintenance as a Stochastic Propagation Process through the graph. We think that the features of XP progress can be well explained by our model.

The Model

A Software System is made all by entities, such as classes, functions, attributes etc., and relations between entities. Thus, it seems natural a graph representation where each entity is a node and each relation is an arc. Let's call our model *RG Model*. The RG Model is made of a set of nodes and a set of relationships (that is, couples of nodes) between them. Each node and each relationship has its type. Five different types of node and nine different types of relationship exist as shown in the following table.

RG Relationship	**Client Type ⇒ Server Type**
Contains	Class ⇒ Attribute Interface ⇒ Attribute
InnerClass	Class ⇒ Class Global ⇒ Class
Function	Class ⇒ Method Interface ⇒ Method
Extands	Class ⇒ Class Intreface ⇒ Interface
Calls	Method ⇒ Method
Uses	Class ⇒ Attribute Method ⇒ Attribute
Parameter	Attribute ⇒ Method
Instance	Attribute ⇒ Class
Implements	Class ⇒ Interface

Each node of the RG Model has the following set of additional attributes:

Name (two different nodes cannot share the same name), *Links* (the list of relationships whose the node is an element), *Value* (different meaning on different context), *Metrics* (a list of metrics).

Each relationship is made of a couple of nodes: one is called *Server* and the other is called *Client*. Furthermore, for each relationship type is given a probability to the server (ProbToServer) and a probability to the client (ProbToClient).

To focus on randomness of maintenance process it could be better to look at the RG model as a Graph rather than a Random Graph: given the system there is no randomness on the topology of his representation. A generic maintenance operation could be seen as a random visit on the RG Model starting from a node, which represents the module needing change. For instance, suppose the node n_i of the RG Model is the representation of the module m_i of the system. Then, suppose a change is needed on m_i. A maintenance operation is represented on the RG Model as a visit staring from n_i. The visit propagates randomly to the n_i connected nodes through the n_i links. The probability of propagation is given by probToServer, whether n_i is the client of the selected link, or by ProbToClient if n_i is the server. The random visit proceeds recursively on the n_i connected nodes and propagate within the graph.

We define four different types of visits. Each visit starts from a node and propagates randomly across its links. Visits are repeated starting from all nodes of type Class within the RG Model.

Each kind of visit produces a specific metric (RG Metric). Thus, four different metrics are defined:

- *Costs:* a cost driver, which takes account of number of lines of code (LOC) visited during the random propagation.
- *Visits:* takes account of the number of times a node of type Class is traversed during whole simulation.
- *Walks:* takes account of the path length crossed starting from a node of type Class, during the whole simulation.
- *Marks:* takes account of the number of nodes touched starting from a node of type Class, during the whole simulation.

Furthermore, we define four RG metrics based global variables, for having a view upon the whole system.