Collaborative Feature Modeling: a Voting Based Approach with Divergence Tolerance

Outline

1. Introduction

Background: feature model in today’s software engineering.

The problem: domain changes too complex and fast changing to be mastered by any domain expert. Furthermore, construction of feature models is tedious and error prone. 难以找到一组领域专家去建模。领域模型的质量难以保证 by 几个领域专家，且缺乏演化

What do we have now cannot tackle this problem.

Our solution: collaborative feature modeling, by capturing the consensus and divergence among modelers to show a whole understanding of the domain, while allowing individuals to benefit from others and express their own understanding of the domain. 简述用什么方式达到上述效果。

We allow users to create their parts of model without the risk of overwriting others’ work.

However, we still allow them to eliminate unacceptable parts from the model, if they can reach such a consensus. Thus, if most users are responsible, the quality of the model can be assured.

We achieve these by allowing them only to create and vote, and implementing other operations via these two operations.

We extend the meta-model of feature model to support our mechanisms. Finally, we have developed a tool prototype to verify our theories.

The structure of this paper ends the introduction.

2. Related Works

Works on generic collaboration: use locking and revisions to prevent overwriting

Take CSCW Seriously: defined a research roadmap for CSCW field.

The important concepts of collaborative system are defined by several papers such as:

Dourish, P.; Bellotti, V. (1992). "Awareness and coordination in shared workspaces”

The success of CSCW systems are often so contingent on the peculiarities of the social context that it is hard to generalize. Consequently, CSCW systems that are based on the design of successful ones may fail to be appropriated in other seemingly similar contexts for a variety of reasons that are nearly impossible to identify a priori

Grudin, J. (1988). "Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces".

Ontolingua Server: a tool for collaborative ontology construction

Farquhar, A | Fikes, R | Rice, J

International Journal of Human-Computers Studies. Vol. 46, no. 6, pp. 707-727. 1997

Wikipedia

Works on collaborative software engineering

Collaborative Software Engineering

A World-Wide-Web Architecture for Collaborative Software Design.

Works on feature model, no one has explicitly introduced “collaboration” in the process of construction, yet there were works on collaborative configuration.

Works on feature model verification (ours)

3. Preliminaries

Feature Model and verification based on our previous work

4. Collaboration Mechanisms

概念模型：加入人的因素

OCL：说明元模型里无法用图形表示的性质和约束

Key points:

[Static]

Global Model, Working Model, Private Model, and their meanings (reflections)

Objects and operations on them (create and vote)

Vote propagation rules

[Dynamic]

Activity-level coordination (the process)

Verification

Feature-level coordination (what happens when multiple users operate on same feature)

Structure:

4.1 Overview of Our Collaboration Mechanism (move to section 1)

4.2 Basic Operations: Create and Vote

4.3 Voting Propagation

First thing is what happens if a user voted ‘NO’ on a feature, or a relationship, or a value of an attribute?

Second, why we didn’t propagate the vote on one feature to another via the requiring/excluding relationships? Because the propagation needs to assume the relationship is correct.

4.3 Global Model, Working Model and Private Model

First, what does each model mean?

Also, we need some automatic propagations when construct PM from WM.

Construction of a private model is optional.

5. The Process

5.1 Overview of a typical process

User joins a modeling project.

Generate a Working model (which equals to global model now) and private model (which is empty) for him.

Verify the working model, report the errors and controversial elements.

User solves the errors and reviews the controversial elements via voting and creates new elements, in his working model.

The working model and global model change along with each user’s operations, including this user himself.

The private model changes along with its owners operations.

5.2 Verification on Working Model and Private Model

Verification on the working model informs the user that there are some works to be done.

New error types: more than one parent, no parent

Verification on the private model reports an abnormal model. We should notice there couldn’t be errors in this model.

6. A Case Study

An introduction to our tool prototype: ways to minimize the possibility of redundant work (create feature by name searching, etc.), highlight the controversial parts.

7. Conclusions and Future Work

Feature Model Evaluation?