
Ontology-based Software Engineering



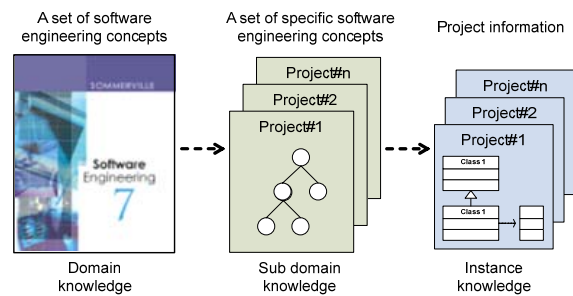
Dr. P Wongthongtham
Digital Ecosystems and Business Intelligence
Institute (DEBI)
www.seontology.org

Overview

- Introduction
 - The Software Engineering
 - Challenges and Limitations
 - Ontology-based Software Engineering Solution
 - Conceptual Framework
 - The Software Engineering Ontology
 - Demonstration
 - Software Engineering Ontology Applications
 - Demonstration
 - Future Work
-

The Software Engineering

- An engineering discipline focusing on the development of high quality software applications – Ian Sommerville
- All aspects of software production from the early stages of specification through to management of software systems



Challenges and Limitations

- Training and practice are different between cities and countries
 - Communication issue over software engineering knowledge
- Software engineering principles and discipline are not understood and followed
 - Inconsistency in presentation, documentation, design and diagrams

Challenges and Limitations

- Inconsistency in understanding software engineering theories and practice
 - When software engineers communicate, their own knowledge base and terminology is different from others



"People cannot share knowledge if they do not speak a common language" - Davenport and Prusak

Ontology-based Solution

- "Ontology is a shared specification of a conceptualisation" - Gruber
 - Ontologies provide a shared and common understanding of a domain
- Ontologies are formal and machine manipulable
 - Ontologies are unstandable for human and machines

Ontology-based Solution

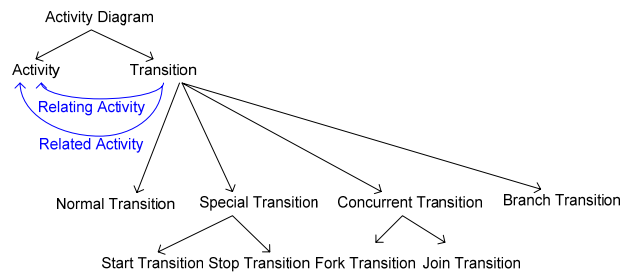
- For our purposes,
 - Software engineering ontology provides software engineering concepts for representing and communicating over software engineering knowledge.
 - Software engineering ontology defines the concepts – what they are, how they are related, and can be related to one another
 - e.g. there are activities and activities have transitions in an UML activity diagram domain

Ontology-based Solution

- Ontology basically has two distinct components:
 - Concepts and its relationships in the domain
 - Activity is a concept in an activity diagram
 - Transition is a concept in an activity diagram that relates between activities that being transited or transformed its state
 - Constraints on the domain
 - Fork transition constraints
 - restricts at least two activities in related activities
 - Only one activity in relating activity

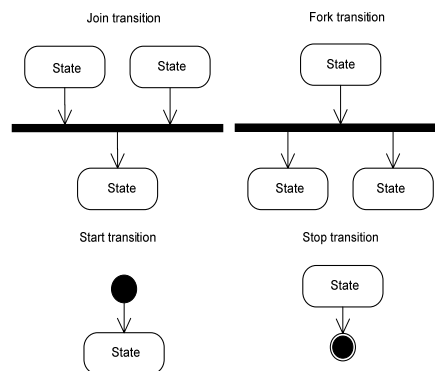
Ontology-based Solution

- Concepts and its relationships in the domain of an activity diagram
 - Activity is a concept in an activity diagram
 - Transition is a concept in an activity diagram that relates between activities that being transited or transformed its state

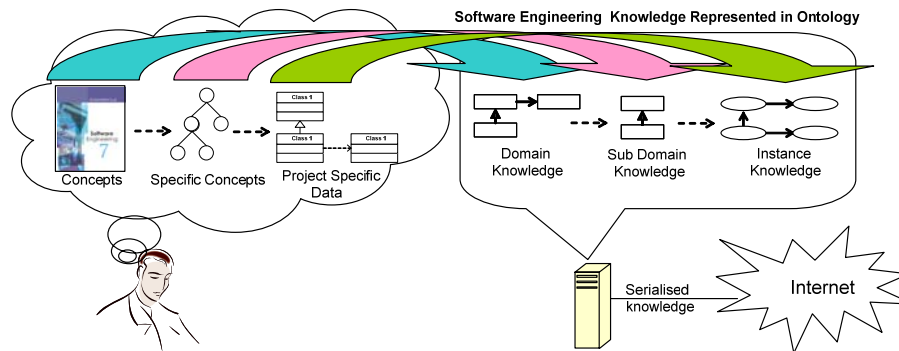


Ontology-based Solution

- Constraints on the domain of an activity diagram
 - Fork transition constraints
 - At least two activities in related activities
 - Only one activity in relating activity
 - Join transition constraints
 - At least two activities in relating activities
 - Only one activity in related activity
 - Start transition constraints
 - At least one activity in related activity
 - No activity in relating activity
 - Stop transition constraints
 - At least one activity in relating activity
 - No activity in related activity



Conceptual Framework



Conceptual Framework

- Why develop the software engineering ontology
 - To define software engineering project resources more precisely and make them more amenable to machine processing as they have evolved
 - To make software engineering domain assumptions explicit
 - To separate domain knowledge and sub domain knowledge from instance knowledge
 - Multi-site software development reference for applications
 - To share a consistent understanding of what software engineering project information means

Conceptual Framework

Software Engineering Ontology

Abstract + Instantiations



- Software engineering knowledge is represented in the software engineering ontology whose instantiations, which are undergoing evolution
 - To reflect project development, changes in the software requirements or in the design process,
 - To incorporate additional functionality to systems,
 - To allow incremental improvement, etc.

The Software Engineering Ontology



Demonstration

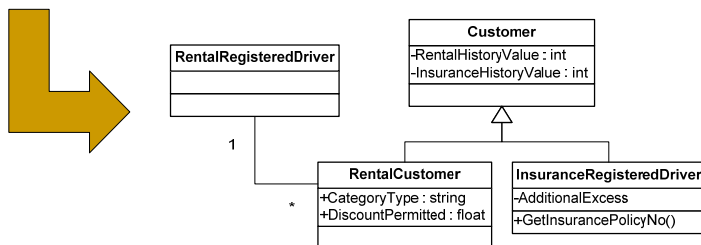
Software Engineering Ontology Applications

- The software engineering ontology
 - Make software engineering project information machine processable
 - Offer prospect of enhanced project information management

Software Engineering Ontology Applications

I am struggling to understand why we need it. I think the system will be simpler for people to understand if we deleted the insurance registered driver.

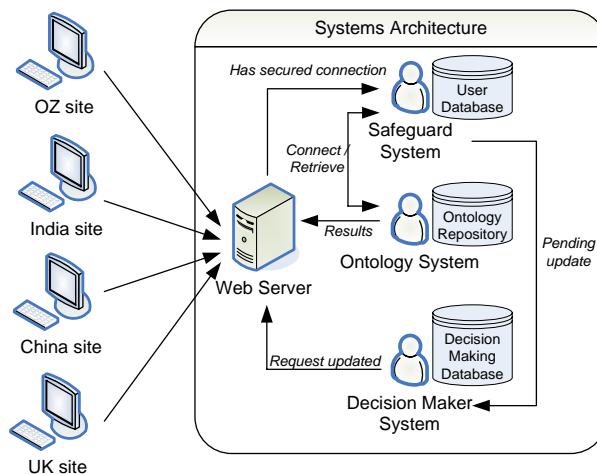
*My reasons for this are that the **insurance registered driver** is a sub type of the **customer**. This means that for every insurance registered driver object there must be a corresponding customer object. However, in the customer object we store values like customer type, insurance history value and rental history value. It does not make sense to have these values for the insurance registered driver. I also think people will be confused because we have the **rental registered driver** as an association with the **rental customer** (which is a sub type of the customer) but the insurance registered driver is a sub type of the customer.*



Software Engineering Ontology Applications

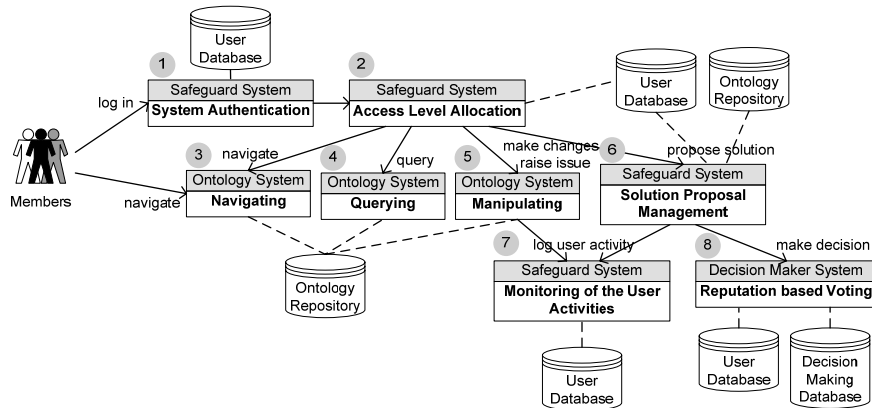
- Data relating to the project will be populated to the software engineering ontology
 - Mapping various project data to the concepts defined in the software engineering ontology
- In a large volume of evolved project information, systematic management is needed!!

Software Engineering Ontology Applications



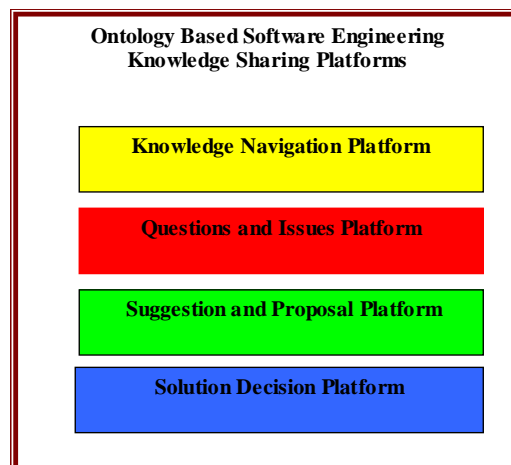
Model of management systems

Software Engineering Ontology Applications



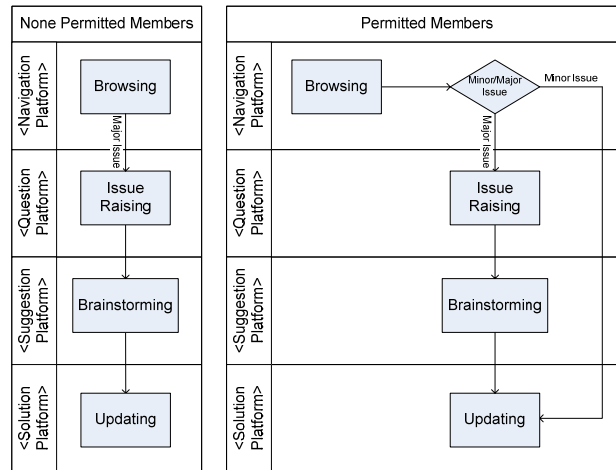
Functionalities provided in each system

Software Engineering Ontology Applications



Software engineering knowledge sharing platforms

Software Engineering Ontology Applications



A flow of the processes when issue arises

Software Engineering Ontology Applications

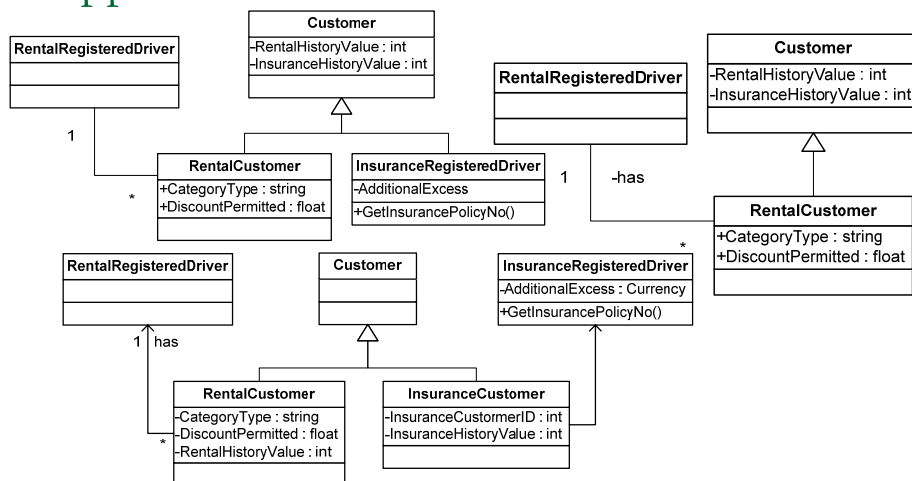
- The decision making is based on members in the teams agreeing to vote
 - The reputation of each individual involved in the software engineering project
 - Using the Markov Model – Chang, Dillon & Hussain 2006
 - Area of expertise
 - “Members who actually work on a task have the best understanding of that task”

Software Engineering Ontology Applications

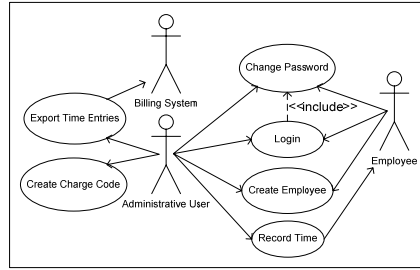
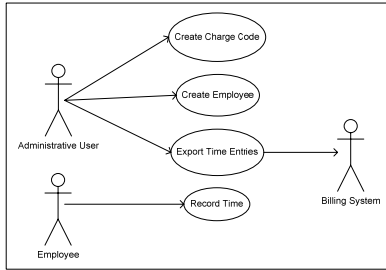
<http://tomasz.ath.cx:8080/seprotege/>

Demonstration

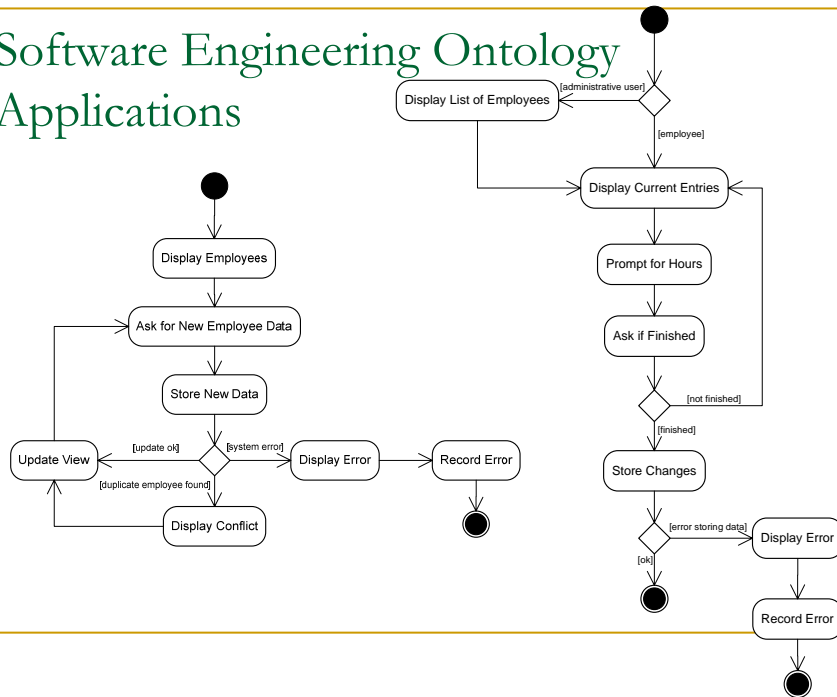
Software Engineering Ontology Applications



Software Engineering Ontology Applications



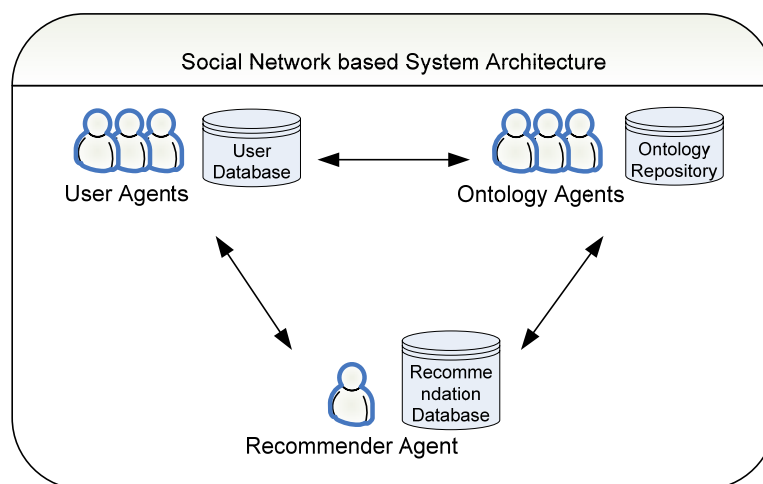
Software Engineering Ontology Applications



Future Work

- “Social Network based Approach for Software Engineering Ontology Sharing and Evolution in Multi-site Environment”
 - Integrates the software engineering ontology and expert recommendation facilities for communities of software developers remotely working on related software engineering projects
 - The software engineering ontology enables an active ecology of SN agents to convey, consume and act on project information (semi) autonomously, according to explicit software engineering domain knowledge
 - Recommendation techniques are addressed to make progress – ability to recommend useful project information, solution(s) for project issues that arise as experts

Future Work



Main interactions among agents

Thanks for your time & attention

Welcome
Questions / Suggestions