

Welcome to Semantic eScience 2011 Fall!

Scribe: Sumitra Madhav

Next week's Scribe: Amar Kannan

==Agenda==

9/26/2011

- * Class Exercise 1: Use Cases Presented by Professor Peter Fox
- * Questions on reading
- * Use Case Introduction and elaboration
- * Use Case development on-the-fly

==Attendance==

- * Peter Fox(pfox@cs.rpi.edu)
- * Weijing Chen(chenw8@rpi.edu)

Registered Students:

- * Amar Kannan(kannaa@rpi.edu)
- * Sumitra Madhav (madhas@rpi.edu)
- * William Gill (gillw2@rpi.edu)
- * Daniel Rotondo (rotond@rpi.edu)
- * Akeem Shirley (shirla@rpi.edu)
- * Apurva Tiwari (tiwara2@rpi.edu)
- * Han Wang (wangh17@rpi.edu)
- * Michael Barron (barrom2@rpi.edu)
- * Linyun Fu (ful2@rpi.edu)
- * Amuruta Akut (akuta@rpi.edu)
- * Daniel Rotondo (rotond@rpi.edu)
- * Apurva Tiwari (tiwara@rpi.edu)
- * David Molik (molikd@rpi.edu)
- * Rohan Dhruva (dhruvr@rpi.edu)
- * Prasad Kathyayani G. (prasak@rpi.edu)
- * Chhaochharia Hithika (chhaoh@rpi.edu)
- * Katie Dunn(dunnk2@rpi.edu)
- * Charisma Ladiwala (ladiwc@rpi.edu)

==Past Action Items==

*

==Action Items==

- * Assignment 2 is up on the course website
- * Assignment 2: Use case Driven Knowledge Encoding Part 1, Part 2 is the presentations. (Due Tuesday Oct 11, Tuesday follows Mondays schedule, presentations due in class 6)

==Notes==

Semantic Web Methodology - Who is involved in using and building the use case

Use Case development team:

- Distinct roles and skill sets
- Ideal team comprise of 7-9 people comprising of:
 - Facilitator has to be aware of the method
 - Domain experts - in particular those who know what resources and data are available

- Modelers - Someone to be looking at object oriented design of the use case description
- Software engineers who are familiar with architecture and technology available
- Scribe to take notes and record comments
- Since it is a team effort, social aspect is key to understand team dynamics
- For assignment 2:
 - Facilitating yourself is something to think about
 - Depending on what use cases you are thinking about, engage domain experts by reaching out and / or share this role
 - You will have to do knowledge modeling, generate some amount of triples to query
 - Although there is no implementation, there is a role of s/w engineering
 - Means of writing as much down as possible - wiki or blog etc
 - Not expected to have a s/w implementation but some extent of knowledge modeling is required
 - Not expected to get the best ontology or perfect encoding
 - While it is not needed to implement the use case, it is required that the use case has reasonable scope and is implementable
 - For Ref: http://wiki.esipfed.org/index.php/SolutionsUseCase_Template :
 - Wiki based representation of a fairly fine level of granularity for capturing use cases
 - Individuals can go in and edit to fine level of detail
 - Has a good versioning mechanism
 - Elements in the section are similar to the short form but not identical
- Use Case:
 - sequence of interactions in a system that you might want to build
 - has to have a goal and is really important to structure the use case to identify and define the goal so that way once you implement the use case, you can check if you have achieved the goal
 - anything that is irrelevant to the actors should not be included in the use case and it is the job of the facilitator / modeler to point out if it is needed or not. If it is external to the use case, there is a section in the use case template where you can note that down
 - Identification of functional and non-functional requirements are also part of developing the use case
 - The main goal of the use case is to be implementable
 - Many implementations of the use case may be highly manual but with the intent of automate it as you move on, working with the end user
 - **Myths about Use Cases:**
 - need lots of use cases (10-100) to build what is needed
 - need of general functionality
 - but in reality you have to have instances to ground it
 - Qn: Is it a bottom up approach? Ans: Only partly
 - need to be an object-oriented design approach
 - have to get them perfect first time
 - only used for s/w development
 - in reality also used for knowlwdge modeling
 - System requirements have to be identified in the use case
 - In the example:
 - The top of the purple line are very top level attributes
 - Hazard planner reports to hazard authority. To achieve the same, access to tools, customize data products etc.
 - This high level design is shown to the hazard planner
 - Below the line is where you drill down to the details
 - Describes how the Agency Specialist achieves his role and performs his responsibilities
 - The Hazard Planner does not have knowledge about most of the technical

details done by the Agency Specialist

- Curator does his/her job based on the information provided by the Planner
- Use Case Examples:
 - Melbourne, Aus: Collection of any data format model run datasets available for internet access with web browsing to find suitable data and access to data via matlab
 - Much too vague
 - Too large a scope
 - Provide browse and quick look access to broad variety of climate, weather and ocean data:
 - Too broad
 - Very subtle goal
 - Install an OpenDap Hyrax Server with THREDDS cataloging on the front-end to support netCDF and ...
 - Very clear about the goals
 - Good technical use case
 - Particular technology choices and architectural specifications that may or may not fit with what you want to develop, much too specific.
 - No semantic capability in any of those technologies
 - High performance data transfer of specific climate model products into the climate diagnostic and analysis tool (CDAT) for analysis, independent of their storage..
 - Does not talk about what the user would get out of this use case
 - Has both functional and non functional requirement(high performance)
 - US 9th grade teacher is preparing a lesson plan aimed at getting students to learn more about the northern lights addressing NSES content standards in earth science. The teacher wants the students to learn the scientific terminology,...
 - Has a goal!
 - Set of preconditions that are vaguely states
 - In addition to the explicit goal there are some sub-goals also stated, this gives an opportunity to do some scoping on this use case
 - Real use cases: Marine habitat-change
 - Particular use of fishing
 - Inter-relations between fish, rocks, salinity etc
 - NEFSC ESR:
 - Implicit is what figures and tables need to be used, what datasets are to be used
 - Non functional requirement-efficient. These type of words should generally not be used in the use case description.
 - Additional component is to look into metrics of some sort to check if efficiency is achieved.
- Use case Template:
 - Name of the use case
 - Point of contact: Yourself
 - Summary- details the business case of the use case, why you want to implement this use case, where it fits in, can talk about the indicators mentioned in the use case, limitations, delegations of roles and responsibilities etc, scalable or non-scalable, usually is a written description of the use case.
 - Actors:
 - Can be verbose here
 - "Person" is not usually considered as an actor although "general public" may be acceptable
 - Triggers
 - Basic Flow:
 - 8-10 lines
 - Good level of granularity representing what needs to be done
- Activity Diagram:

- Any diagram with a good level of detail about actors, roles etc will do
- Use case format:
 - Short format is good for the exploratory phase of the project, good for including a proposal and for activities that lead up to the actual implementation
 - Name and goal: Concise, enough to be recognizable, avoid jargon or acronyms, can be the same as the use case name but often is not, state goal concisely (can start with "goal of this use case is"), update the goal to mesh with the summary
 - Scoping: focus on core functionality, must resist tendency to generalise the use case, acknowledge other important issues such as required vs. optional, non-functional requirement, available personnel skills and resources
 - Summary: MOST important part of the use case, state business case, background, goal in detail, success and failure scenarios, actors and their roles, describe the "how" of the functionality, describe a successful outcome
 - Triggers: Conditions that initiate the use case, that is prior to the first step in the normal flow. Something that appears here SHOULD NOT appear in the normal flow of the use case. Can be triggered by things outside and / or inside the use case. Often start with 1 trigger and can think of adding others later.
 - For Ex: Recording a hazard event: if the general public calls up to find out / report whats going on, this may not be within the scope of the use case, but is a trigger.
 - Actors: Initial analysis will have many human actors often, but think where they can be replaced where they can be replaced by machine actors. If this is done in a team, think of the service-oriented paradigm i.e think of inputs, outputs and preconditions for each actor.
 - Two types:
 - Real People and Blockheads (computers)
 - Real people can be replaced by blockheads in reality
 - Primary actors initiate / requests and secondary actors repond / act upon.
 - Ex: Analyst(primary actor) interacts with an interface (secondary actor)
 - Ex: Hazard planner(primary actor) acts on the reports for the hazard authority(secondary actor)
 - Precondition: All conditions that are true for the trigger to cause initiation of the use case, i.e requirements. Have to be in existence before use case is executed. Some level of modeling of preconditions may be required, beware of using other entities data and services. Attempt and succeed in filling in complete lines in the table about data /models / events/ applications.
 - Post condition: Change in state before and after the work flow is executed.
 - Ex: where the data products may be located
 - Success and failure scenarios: To be included in the summary. All details of when you did not achieve a certain goal, whether the failure was due to lack of information, any impact on metrics etc must be included
 - Normal Process Flows: Set of steps after use case is triggered, steps are separated by actors, idea is to modularize the part of the work flow, can have loops. Can use the activitey diagram as a means to turn the written process flow into a visual one or vice versa. Don't try to make the activity diagram into a flowchart!
 - Alternate Process Flows: Variations that can be invoked by valid but not necessarily useful logic. Don't usually represent error conditions. Usually put 1 in the document and the rest in the notes.
 - Ex: In the eco system report, the status report was machine generated instead of manually
 - Non-Functional Requirements: Define HOW a system is supposed to be. Before that make sure what the WHAT is (i.e Functional Requirements). Two Types:
 - Execution and evolution qualities
 - Artifacts: Use case generate a number of things. Ex: Log, graphics, provenance information etc. It is oftern useful to record those that are important and name them, give

them a URI, make sure to record what went into making them. Often useful to have the actors name them and put them in a place as they are responsible for the artifacts. Keep in mind that when you want to pull local-data, you have access to some sort of a sandbox or any other resource from where you can get the information you need for implementation. That is why it is important to review and document every resource.

- Have an elevator pitch ready for why you chose this use case!
- Resources:
 - All resources are listed on the course website
 - Best book on use case : Bittner and Spence, published 2002 or 2005, "Use Case Design"
 - Version 5 of Omnigraffle for Mac
 - Cmap : <http://cmap.ihmc.us/> version 4.11.x or 4.18.x (use /coe at the end of the link for cmap ontology editor)
- NOTES:
 - Just get the narrative description of the actors the first time you design the use case individually
 - Draw really simple diagrams : UML4US
 - Hints:
 - Write down the name and the goal and start on the summary
 - Go back and review the goal
 - List the actors, preconditions and triggers
 - Go back and review the summary to see if these are well described
 - Go back and review the goal
 - Review actors, preconditions and triggers and so on

Break for 5 minutes

Use case class exercise

1. Find probability of crime occurring on a particular block

- What conditions are required for this to occur? (location, date, what the person was wearing-- profiling, history of crime in that area)
- Which city? between x and y streets, M & N avenue, Seattle Washington - you get a better understanding of what the concept of a block is.
- What is the goal? To make a decision regarding safety for living
- Now the goal looks like : Make a decision regarding suitability and safety of living in this ^ block of Seattle. Details should be elaborated in the summary.
- Distinguish between the past data and what is going to happen in the future and provide an interface to the past data?
- If your criteria is probability then it is important to articulate the decision. (if probability is 1, then what? if not, then what?)
- What is a crime? Is it a felony crime? Would you predefine the definition of crime or would you look up a standard definition?
- What are the first class objects, resources and concepts you need to represent? --Types of crimes, features of individuals who commit the crime--> (another actor: perpetrator of the crime which is at the moment irrelevant, but can be included in the notes) This forces us to look at the boundaries of the use case.
- What time are you identifying? Daytime vs. nighttime?
- Example of modeling : Crime occurs at location at a particular time limited by understanding of the victim. Other things are secondary, supporting that and not a high level object in the use case.
- Actors in this use case: You (a person of a certain age), data provider--> police or other provider of crime statistics (secondary actor), etc
- Trigger: Need to make a decision regarding a potential job offer in Seattle, relocation decision etc.

2. Within a given area, number of health insurance agencies that have support for diabetes

- What area? New York city, Bronx
- Goal is not to accommodate all the people with diabetes in Bronx but only 1 individual.
- Goal: Give an individual all the required health information about diabetes health insurance agencies
- The individual may travel within or outside an area. So look at where the insurance agencies may operate.
- Are agencies the same as providers or plans or brokers? They are all distinct but support one another.
- Does the insurance cover all types of diabetes? People who buy the insurance vs people who don't buy it?
- Support --> most likely the insurance companies only list what their diabetes support look like in text as opposed to a web service?
- An advantage of choosing a use case that revolves around your life is that you have some level of domain expertise.

Qn: What level of granularity must be expressed in the goal?

Ans: You can never be too specific. But keep in mind the broader goal of the use case. There is no magic formula!

3. How are search queries evaluated in a search engine

- The goal is to understand how a search engine works possibly to retrieve better results
 - What does understanding mean? -- How it works
 - What sort of things to be understood? -- Query structure, how queries are filtered, how semantics are implemented, ranking algorithm
 - Is there any comparison with search engine x vs. search engine y.?
 - Goal : Better understanding of how search works to achieve better results
-
- Summary:
 - Use cases are a powerful tool
 - Should drive the functional requirements of both your ontology and how you build one
 - Reusing parts of other people's ontologies if possible