NASA: Trust-Based Data Service Reuse and Recommendation

Task 1: Planning roadmap

Version 2.0

Team NASA:

- Neeraj Saini (Project Manager)
- Pujita Rao
- Shuai Wang
- Venkatesh Sriram

Faculty Advisor:

- Jia Zhang
- Martin Griss

Submitted on: December 6, 2013

Executive summary

This document gives a general description of the "Trust-Based Data Service Reuse and Recommendation" project. In the following pages we provide a final version of our roadmap, which we have revised during the course of the project, in multiple iterations. This project was intended to develop tools that would eventually provide a recommendation technique that enables efficient reuse of scientific work to facilitate scientific experiment design.

In this document, we give an introduction of the project, listing its goals and stakeholders, and provide the schedule of our project. It shows how we have followed the target dates we had set for ourselves for the completion of iterations and how we had kept track of deadlines for deliverables. Also included in this document, are the Agile practices (SCRUM techniques) that our team planned and implemented. It also states the high-level organizational decisions that we made to secure the success of the project (i.e., code review). In addition to this, it also lists the constraints given by the context of the project.

Finally, some conclusions are presented in this document, based on the changes we made to the design. This is a final version of our planning roadmap, which we followed to complete this project.

The project

The "Trust-Based Data Service Reuse and Recommendation" project intends to develop a trust-based recommendation engine for scientists. The engine will recommend existing reusable artifacts (e.g., analytical tools, existing models and workflows) hosted at NEX central Database. Related papers may be recommended as well. The initial phase will be dedicated to resuming work from last year. After this we would build a REST API and integrate this with NASA's VisTrails tool. To enhance the system we would provide recommendations of researchers to collaborate with. These recommendations will be improved by adding additional features and implementing them using various machine learning algorithms over multiple iterations.

Stakeholders

The team has identified all the stakeholders of this practicum project:

- **Client:** Dr. Ramakrishna Nemani, Petr Votava @ NASA
- **Project Advisors:** Jia Zhang, Martin Griss
- **Project Team:** Neeraj Saini, Pujita Rao, Shuai Wang, Venkatesh Sriram

Effort: All team members will work on the development and share responsibilities. Since there is no solid line dividing the front-end and back-end, allocation of work has not been done based on layers of the project. Instead, team members will work together on assigned user stories.

Goals and purpose of the project

The broader goal of this project is to build an intelligent recommendation engine based on human trust where scientists can search and get most relevant results in their domain. The idea behind this project is to help scientists by providing most relevant information from the existing knowledge based to avoid from duplicating efforts and eventually increase their work efficiency in their daily scientific work.

Within the scope of this project, our major objectives to be accomplished are:
1. Conduct a feasibility study to explore whether hidden knowledge can be extracted from social networks to support and facilitate software service discovery
2. Propose a context-aware software trust model
3. Develop, as a proof of concept, a prototyping system of software recommendation engine that calculates probability of co-authorship.

The initial goal is to get the last year's team's code up and running and integrate the existing work into Vistrails as a plug-in.

Possible deliverables and timelines

The team has identified the following deliverables:

1) Working algorithm and engine should be implemented
2) Documentation: The deliverables include the following documents:
    - Technical Report
    - Design Document
    - User Manual

This documentation will help the following work after the Practicum Presentation. The team will prepare slides for presentation and a demo of this project at the end of this practicum project.

Schedule

Iteration #	Time	Tasks	Deliverables
Iteration 0	2 weeks (Aug 26th to Sept 9th)	• Understand project goals • Identify stakeholders • Identify team roles • Gather requirements	• Engineering Notebook v0.1 • Roadmap v0.1
Iteration 1	3 weeks (Sept 10th to Sept 30th)	• Set up the environment and recover and resume last year's work • Gather feedback from stakeholders. • Update Documents	• Engineering Notebook v0.2 • Practicum Planning Roadmap v1 (Sept 18th) • Product v1
Iteration 2	2 weeks (Oct 1st to Oct 14th)	• Set up REST interface • Add new features • Continuous integration and testing. • Update documents	• Engineering Notebook v0.3 • Product v2 • Individual Reflection Proposal (Oct 13th) • Statement of Work (Oct 6th)
Iteration 3	3 weeks (Oct 15th to Nov 4th)	• Code Review • Integrate into VisTrails with Chris • Add new features. • Continuous integration and testing. • Collect client feedback • Update documents	• Engineering Notebook v0.4 • Technical Report v0.1 • Product v3
Iteration 4	3 weeks (Nov 5th to Nov 25th)	• Add new features. • Continuous integration and testing. • Update documents.	• Engineering Notebook v0.5 • Technical Report v0.2 • Product v4
Final Iteration	2 weeks (Nov 26th to Dec 2nd - 6th)	• Code review. • Finish documentation. • Prepare presentation slides. • Review deliverables. • Prepare for demo and presentation.	• Slides • Product v5 • Technical Report v0.3 • Roadmap v2 • Individual Reflection Report (Dec 6th)

The methodology and approach

Methodology Selection

Based on the nature of the project, our team has decided that the Agile methodology based on Scrum will be the best fit for the project. This decision is made based mainly on the following conditions:

1) Team:
Scrum is most efficient when team has six members or less. Team NASA consists of four members which Scrum can handle fairly well. Also, all team members have experience of working under the Agile methodology in the software engineering course last year, which indicates that the team can adapt to Scrum without needing any extra effort for training.

2) Customer Involvement:
Based on the previous meetings, we think that our customer don't have specific test cases and fully prepared requirements, so we need customers to provide clarification and feedback as the project progress. Common development methodology, like waterfall model doesn't fit in this situation. While scrum encourages active stakeholder participation continually at all stages throughout the development lifecycle. For the NASA project, it is not possible for the customer to work with the team on-site all the time. However, our customer is locally based in the Ames Research Park, a weekly meeting will be set to communicate with the customer. Also, interaction with NASA's domain scientist is also needed for building the trust model. For this reason, the customer involvement is high and meets the requirement of Agile process.

3) Requirements Change:
Coping with changes in software requirements is a very significant consideration when selecting a software methodology. Scrum is more appropriate for projects with frequent and unpredictable requirements' change. The team expects that most of the requirements will be discovered along the lifecycle of the project, especially on the trust model and service expectation. The high level of change in the requirements makes Scrum to be the best solution to handle this situation.

Approach

The process will be divided into four two/three-week development iterations and Iteration 0. A few actions are defined to be done in each of these iterations.

1) Iteration 0 - 1

a. **Visioning:** Since a practicum team of last year has done some work on trust model and data analysis, our team will start with understanding what we currently have. In order to have a better understanding on the concept of "trust model", some background research will be conducted. Once the team has some preliminary background on it, user scenarios will be created to elicit the requirements. A meeting with domain scientist and NASA computer engineer will also help verify whether the team's vision conforms to the values of these stakeholders. After the meeting, our team will come up with some user scenarios and discuss with advisors. Then we will be able to choose the most suitable goals for this team and decide how we are going to approach the challenges.

b. **Backlog and user stories:** The features will be converted to user stories and saved in the Product Backlog. We will be using Asana as our project management tool. The user stories will be prioritized and estimated under the agreement from both the team and the customers. In each of these iterations, we will focus on one part of the main module.

c. **Environment preparation:** The team will schedule regular meetings with the customer and set the tools for testing, coding, integrating and building. The team will use git as the version control and integration tool and Eclipse for Java backend development. For data analysis and storage, mysql will be used and for efficiency SAP HANA database may be used at a later stage.

2) Iteration 2 – 5

a. **Team Meeting:** There are two types of meetings that our team will have each week.
   i. Internal meeting: We will discuss what use cases we want to finish and how we are going to approach it.
   ii. External meeting: We will first show our progress and use cases finished this iteration to customer, then we will discuss with our customer on the use cases they expect during the

next iteration. Also, we will try to elicit and refine user requirement gradually.

b. **Programming and Code Review:** The team will not use Test Driven Development since the requirements can change very frequently and cause extra time to refactor the testing code. And team members will not code in pairs, in order to save the time in implementation and due to schedule constraints. Instead, the team will conduct peer code review to ensure the code quality. Also before starting the implementation, team members will discuss the challenges and proposed solution.

c. **Individual and Team Reflection:** Since this practicum serves as a core course, we will be gathering thoughts on what we learn during and after this project. Apart from the engineering book or final report, we will reflect on the development process.

Constraints and Restrictions

As with any project, we cannot hope to have everything perfect. There are some necessary constraints that we have identified, which may affect the overall development process. These constraints and restrictions include:

1) Continuation of previous team's work:
Since this is a project that is a continuation of what was done by a different team of people, a year back, it will take time for us to understand exactly what their project does, and the design decisions they had undertaken to reach their end result. Thus, before we can use their results and their application, we need to spend time understanding it.

2) Working with an external customer:
We are working to solve a problem for NASA scientists. Thus, while we have their requirements, we still need their constant feedback and suggestions to solve the problem properly. This may not be always possible because they may not always have the time to spare.

3) Requirement Instability:
As an experimenting project, the requirements are not definitely clear at the initial stage of the project. It can be predicted that a lot of new features will be

explored and added into the backlog. For this reason, building a frequent connection channel with the customer is necessary.

Conclusions

After creating the final version of this document, these are our understanding about the initial and final planning of NASA project.

1) Change of Requests: It was clear that the project will have many 'iterations' as time goes on. We needed to perform analysis on data by applying various metrics, and we needed to ensure that we tweak our analysis based on the feedback of our client, who will be the final users of our tool. Thus, it was conceivable that we will have to keep changing our tool as time goes on. By the end of our project, we had considerably fine-tuned our project such that our client was satisfied.

2) User-friendly Interface: Our tool will not be used only by Computer scientists. Instead, we target on domain scientists who may not be familiar with computer science fundamentals. Thus, our tool needed to be intuitive to use and must return meaningful results. For this, we provide visualization graphs and have designed a Web interface that received good feedback form our client.

3) Agile software engineering methodology will be adopted. We needed to slowly build up the system and enhance its capabilities. A good place to start was to build a system that takes a publication as an input, and based on knowledge and trust metrics, gives a thumbs-up or thumbs down to the tool. But instead, we have provided the percentage representing the likeliness to collaborate. It even has details about the features that contributed towards the result. This is an improvement as we would now be telling a researcher something like " The probability of you collaborating with so and so author is 60% because you have a high co-authorship history, medium similarity of research area and low author connectedness". Instead of giving a yes or no answer.