# Simulation-based Communication Tool: an Enabler for Collaborative Decision Making

Paul Stynes[1], Declan O Sullivan[2], Owen Conlan[2]

[1]National College of Ireland, Dublin, Ireland
pstynes@ncirl.ie
[2]University of Dublin, Trinity College, Dublin, Ireland
{declan.osullivan, owen.conlan}@cs.tcd.ie

**Abstract.** Successfully communicating an organisation's goals and rules in such a manner that their impact on IT support can be analysed and discussed, presents a major challenge. The most significant problem is relating the changes desired in a semantically consistent and understandable manner and then reflecting the potential impact of those changes on the organisational structure and the business processes carried out within that organisation. This research investigates a simulation-based communication tool that employs a semantically driven natural-language component to capture an organisation's needs. These needs are translated into a simulation, through the use of semantic web services, of a business process that represent the evolution of the organization's IT infrastructure and policies. Through an iterative communication loop between the business executive, IT architect and staff, the aim of the simulation is to accurately represent and specify the IT changes necessary to support the organisation's needs. An initial scenario-based evaluation of this idea that we have undertaken indicates a desire among executives for such a simulation tool. This paper introduces the problem, architecture and case study, followed by a discussion on an evaluation that is in progress at the National College of Ireland with results expected by the end of March.

## Introduction

Successfully communicating an organisation's goals and rules in such a manner that their potential impact on IT support can be analysed and discussed, presents a major challenge. The most significant problem is in relating the desired changes in a semantically consistent and understandable manner and then reflecting the potential impact of those changes on the organisational structure and the business processes carried out within that organisation.

The aim of this research is to investigate to what extent a simulation tool, based on semantic web services, aids communication in developing a shared representation of IT support changes. This shared representation can aid the necessary collaborative decision making in support of the organisation's goals. An initial scenario-based evaluation indicates a desire among executives' in higher education for such a tool.

This paper presents an approach that aids communication of the organisations needs between different stakeholders such as the business executive, IT architect and staff by allowing each to communicate in a format that they are familiar with using controlled natural language. In addition the paper describes the architecture of the Simulation-based Communication Tool (SCT) that implements the approach to aid communication. Finally there is a section on how this tool will be evaluated among executives, IT architects and staff in higher education at the National College of Ireland.

## Architecture and Case Study

The Simulation-based Communication Tool (SCT) employs a semantically driven natural-language component to capture an organisation's needs through a collaborative communication loop between the business executive, IT architect and staff. The benefit of this approach is that the executive and staff can communicate in a format that they are familiar with through natural language that is semantically controlled by the vocabulary they use within the organisation. Using semantic web services these needs are automatically translated into a simulation of a business process that represents the evolution of the organisation's policies and IT infrastructure. The IT infrastructure changes are in a format that is understandable to an IT architect. This helps to reduce the mismatch in communication between executives and IT architects. The following paragraphs describe the architecture and case study.

The Knowledge Engineer creates an ontological domain model based on concepts and relationships that relate to the organisational goals, rules, structure, business processes and corresponding data. The ontology is documented using a tool such as Protégé [5] which consists of individuals, properties and classes. Individuals represent objects in a domain of interest. Properties link two individuals together. OWL classes are sets that contain individuals. The recommend notation for documenting individuals, properties and classes is the CamelBack notation where names start with a capital letter and do not contain spaces.

The SCT component is responsible for starting up the system and co-ordinates information from the semantically controlled natural language interface.

The Grammar Composition component creates a lexicon from the ontological domain model where individuals are mapped to proper nouns, properties are mapped to verbs and classes are mapped to nouns. The SCT component presents the business executive with the semantically controlled natural language interface. The interface will allow the business executive enter natural language statements using the format "noun – verb – noun.". The controlled natural language is restricted based on concepts from the ontological domain model, to allow reliable semantic analysis of the language.

The business executive creates or modifies the rules for organisational change through the predictive editor, which is part of ACEWiki [3]. The ACE Parsing Engine (APE) [4] ensures that the grammar for the rules is based on a subset of natural language, namely Attempto Controlled English (ACE) [2]. In addition, APE converts the rules to OWL. This results in machine accessible semantics that are automatically processable. The approach with the predictive editor and APE show the potential for natural language interfaces to utilise semantic inference to refine and hone the executive's rules. The predictive editor and APE components represent the only 3$^{rd}$ party components in this architecture.

Validation of the rules are carried out to ensure that the rules are unique and the data in the rule represents valid statements in the ontological domain model. The tool then stores the rules in the ontological domain model.
Simulation plays an important and inexpensive role in portraying the envisaged impact of certain decisions and changes on an organisation or process. Chandrasekaran et al. in [1] discuss an approach for the creation of a simulation model based on web services. This approach forms the basis of the simulation platform in this communication tool. The ontology2ws translator component parses the ontological domain model and applies a template rule to create the web services and simulation data. Deployment of the web services results in the creation of the simulation platform. Staff may also retrieve, create, delete and modify simulation data through the web service interfaces. The innovation with the Simulation-based Communication Tool lies in the collaboration with the IT Architect in creating a shared representation of the evolving IT infrastructure changes that support organisational change through the creation of the web service architecture and data structures during the simulation. The advantage of this approach is the high fidelity between the simulation and real world. It provides an ability to plug real web services into simulated entities, thus creating simulations that utilise as much 'real world' data as possible.
The following scenario illustrates how this system would be used in practice. A knowledge engineer creates an ontology of a higher educational organisation where concepts such as **CourseDirector**, **Lecturer**, **Module**, **Classroom** and **Computers**, are represented as ontology classes in Protégé using the CamelBack notation. Individuals that are part of the ontology classes are also created such as **JohnSmyth**, **DistributedApplicationDevelopment**, **Room3.01**, and **Computer3** respectively. Properties that link two individuals together are created such as **teaches** with a domain **Lecturer** and range **Module**. Examples of other properties are **contains** with a domain **Classroom** and a range **Computers**, **require** with a domain **Module** and a range **Computer**. The knowledge engineer can create Semantic Web Rule Language (SWRL) rules such as "Classroom(?x)∧contains(?x, Computers) ∧Modules(?y) ∧require(?y, Computers) → schedule(?y, ? x)".
An executive, such as a head of school, specifies the organisational needs in english through the controlled natural language interface. An example of such a need is "**JohnSmyth teaches DistributedApplicationDevleopment**.". The tool, performs checks on the statement to ensure that it is a subset of ACE, transforms the rule into RDF and adds the rule to the ontological domain model.
The ontology2ws translator creates web services for the ontology classes **CourseDirector**, **Lecturer**, **Classroom**, **Module** and **Computer**. The object properties are mapped to operations in the web service of their respective domain ontology class. For example JohnSmyth is an individual in the ontology class CourseDirector. This results in a web service for CourseDirector with an operation teaches. An Academic may access the CourseDirector web service and enter the name "JohnSmyth". The web service retrieves operation and parameter data from the database such as, "teaches DistributedApplicationDevelopment. A facilities manager can see what is in each room via the Classroom web service. An IT architect can integrate the deployed web service architecture that supports the organisational change with their current systems. As an example all colleges have a scheduling

system that may not necessarily support collaboration. The IT Architect could integrate the CourseDirector web service with the scheduling system to allow an academic choose their preferred times or classrooms for teaching a module.

## Evaluation

An empirical study at the National College of Ireland will evaluate the communication in developing a shared representation of organisational changes that are needed and can aid the necessary collaborative decision making in support of the organisation's goals between the business executive, and staff through the Semantic-based communication tool. The measures for communication are frequency and quality, where the frequency of communication is the number of messages and responses that take place between business executive and staff. The quality of communication will be measured through a self-report communication log that classifies the communication as (a) Messages that suggest solutions (b) Messages that identify constraints (c) Messages that identify organisational issues (d) Messages that represent collaborative decisions. Effectiveness of operational communication relates to the executive and staffs perception that communication is (a) worse, (b) no different, (c) improved. The operational measure for shared representation relates to the business executive and staffs perception that the simulation based on the web service architecture supports organisational change (a) faithfully, (b) less-than-faithful. Examination of the validity of the tool involves a group consisting of the head of school and the IT manager. Their objective is to rate the solutions based on the measures above.

The study will determine if the tool aids communication in developing a shared representation of IT support changes that are needed and can aid the necessary collaborative decision making in support of the organisation's goals.

## References


1. Chandrasekaran, S., Silver, G., Miller, J., Cardoso, J., Sheth, A. (2002) Web Service Technologies and their synergy with simulation, IN: Proceedings of the 2002 Winter Simulation Conference, 8-11 Dec, San Diego, California. pp. 606-615.
2. Norbert E. Fuchs, Kaarel Kaljurand, and Tobias Kuhn. Attempto Controlled English for Knowledge Representation. In Cristina Baroglio, Piero A. Bonatti, Jan Maluszynski, Massimo Marchiori, Axel Polleres, and Sebastian Schaffert, editors, Reasoning Web, Fourth International Summer School 2008, number 5224 in Lecture Notes in Computer Science, pages 104–124. Springer, 2008.
3. Tobias Kuhn. Combining Semantic Wikis and Controlled Natural Language. Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), CEUR Workshop Proceedings, 2008.
4. Kaarel Kaljurand and Norbert E. Fuchs. Bidirectional mapping between OWL DL and Attempto Controlled English. In Fourth Workshop on Principles and Practice of Semantic Web Reasoning, Budva, Montenegro, 2006.
5. Protégé 3.3 ontology editor. Available from: http://protege.stanford.edu/ [Accessed on 12th March 2009]