Model Unification in Support of Political Process

Peter Weinstein

Altarum Institute
3520 Green Court, Suite 300
Ann Arbor, MI USA 48105
peter.weinstein@altarum.org

Abstract

We are developing techniques for unifying models of business processes that span and integrate organizations. In this paper, we describe our approach and discuss its potential for broader application to decision-making and political process. Key elements of the approach include the following. Participants work with familiar diagrams that the system represents internally in OWL. The system unifies models of analogous processes in a manner where users contribute confirmation and guidance but are not required to provide any particular input. Unified models can be compared, yielding visualizations that crystallize insight and metrics for quantifying alignment. The unification process thus identifies commonality and analyzes differences. This process is therefore well suited to facilitate thorough discussion of policy issues in a manner that fosters creative negotiation and consensus building.

Introduction

Business process modeling has become an important tool for government planners as they work to improve their organizations. Unfortunately, in a cross-organizational context business process models often fail to deliver meaningful insights because models developed by different teams are hard to compare. Modelers use different terminology and styles and this hides genuine differences in the processes [2].

Our Business Process Interoperability Living Ontologies (BPILO) project is developing an alternative approach to modeling business processes that spans and integrates organizations to support decisions about interoperability. The first section of this paper describes this work.

The second section of this paper discusses the extension of our work on business process interoperability to support other types of decisions: namely, the political process itself in which (ideally) participants compare viewpoints, engage in creative negotiation, and emerge with well-considered decisions that further the well-being of society.

Unification of Business Process Models

This section describes how we represent models of business processes, unify them, and compare these models to support decisions about interoperability.

Model Representation

A core element of the BPILO approach is that users work with diagrams and other tools with which they are familiar. The system represents these models internally in OWL. We do not ask users to directly specify formal ontological models because we find that people who lack training for this kind of work often flounder. As a starting point for the BPILO project, we decided to work with business process models in a format familiar to practitioners in the domain. Rectangles represent processes connected by data flows, within swim lanes that identify the groups responsible for each process. Figure 1 shows a snippet from a utility implemented with Microsoft Visio. Models can be exported and imported into the BPILO system, which is implemented within Protégé-OWL.

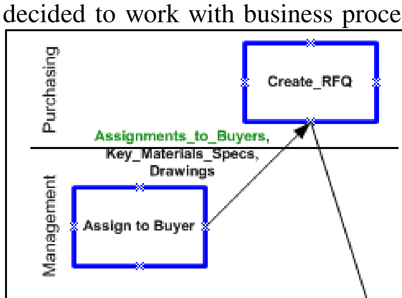


Figure 1: Fragment of a business process model

BPILO's model import capability uses a pre-defined core ontological model of the semantics of business process diagrams. In this model, for example, Tasks are defined as Activities that have input and output data, while Processes are Tasks that can have subtasks. By assuming that all users use our modeling tool, therefore, we start with some degree of ontological commonality. The core model must be constructed carefully but the exact particulars do not matter. The process of converting the user diagrams into ontological models essentially turns meaning that is implicit in the diagrams into explicit relations that are amenable to computational reasoning.

Model Unification

We call models that are freshly imported into BPILO *original models*. Original models contain heterogeneous terminology – such that word meanings across models are inconsistent, redundant, or more generally, overlapping in ways that can be subtle and difficult to reveal. More precisely, original models have two layers as illustrated in Figure 2, where heterogeneous organization-specific concepts inherit some properties from core concepts that are shared (blue layers are shared, the red is not).

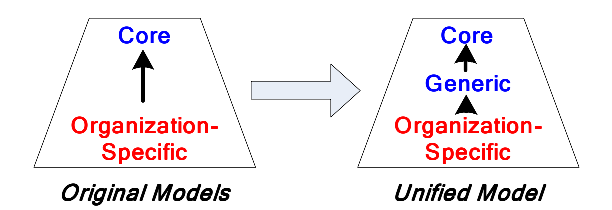


Figure 2: From original models to unified models

Model unification converts original models to *unified models* by creating a shared middle layer of generic concepts from which organization-specific concepts inherit. In the unified models, properties associated with organization-specific concepts represent differences, while properties inherited from the generic layer represent similarities.

We implement model unification with an algorithm in the style of swarm intelligence, inspired by social insects such as ants and wasps, where organization emerges from simple local interactions [3]. Our strategy is to design agents that play roles defined by their associated roles in the unified models. For example, the role of a generic concept is to find the specific concepts that are best matched so as to inherit as much as possible from the generic concept. The swarming unification algorithm has three concurrent processes involving generalization, aggregation, and re-writing of organization-specific terms. So far we have implemented generalization, which matches concepts from original models to create generic concepts.

We describe the generalization algorithm with an analogy to the game of musical chairs. This is because only one concept from each original model is allowed into a match. For example, in Figure 3 the C:Prepare_RFQ concept (from organization C) has asked to move into the match defined by the generic concept G:Create_RFQ. This involves kicking the C:Create_PO concept out of that match.

The Match Agents in the generalization process base their decisions on two kinds of input. We estimate lexical association using the Semantic Lexicon tool from Fair Isaac [1]. These estimates are based on word co-occurrence in a corpus of documents describing business processes. We also estimate structural compatibility: organization-specific concepts are considered more similar if in the original models they are related to concepts that are also matched in the generalization process. Structural relations are also used to accumulate suggestions to agents regarding which matches to join. There is therefore a positive feedback effect, as is typical of swarm intelligence algorithms, where structural isomorphism encourages further matches that express that isomorphism.

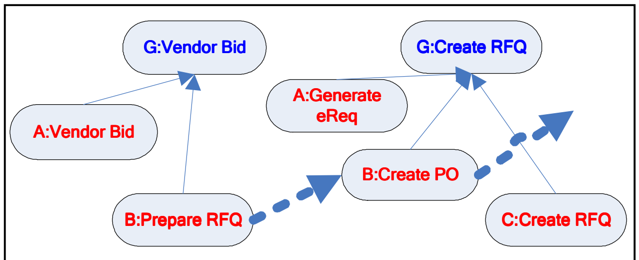


Figure 3: Concept matching as musical chairs

One of the benefits of swarming is that these algorithms tend to adapt gracefully to runtime change in the problem. This supports a style of user interaction that is *anytime* and *anywhere*. Unification proceeds automatically, with modelers contributing guidance and confirmation. The system exposes its processes to users and invites advice, rather than waiting for user inputs that may be difficult to provide.

Model Comparison

The motivating goal of model unification is to support model comparison. This can be achieved with graph matching algorithms that may or may not be swarming in nature [4]. Figure 4 shows a visualization of a comparison of two processes, where one process is drawn in blue, another in green, and the two diagrams are overlaid (this figure is best viewed in color). Aspects of the concept definitions that are modeled in the generic layer are shown in pink. The pink therefore conveys similarity, while blue and green are differences.

Each match and overall comparisons are quantified with a metric that describes the degree of commonality (based on a weighted combination of lexical association and structural isomorphism). In any pragmatic context, however, some differences are more important than others. Therefore, we anticipate developing a variety of domain-specific metrics that reflect the objectives of analysis in particular situations.

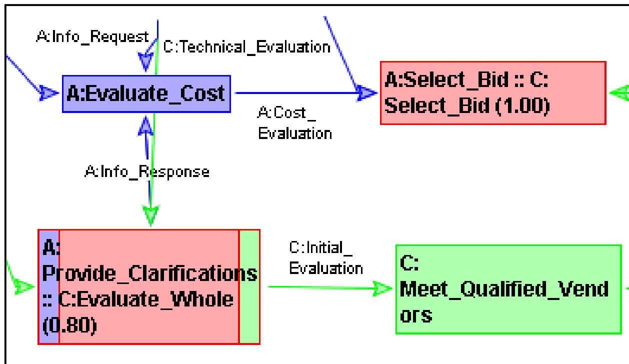


Figure 4: Fragment of a process comparison visualization

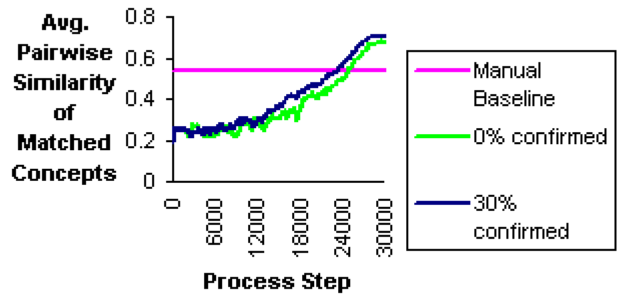


Figure 5: Identifying commonality

Figure 5 shows the average degree of commonality achieved in experiments that include simulated user confirmations of concept matches [6]. The horizontal baseline reflects a unified model that was manually generated. The fact that the system achieves more commonality than is present in the manual model means we need to capture more information in the input models.

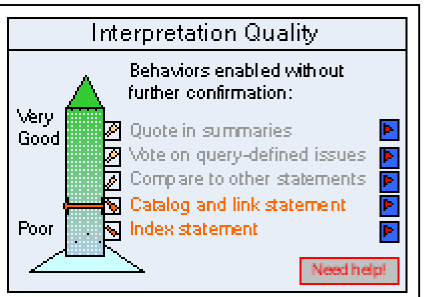
Consensus Builder

Consensus Builder is a vision for an internet application that will organize and facilitate political dialog involving large numbers of participants [5]. People will come to Consensus Builder to:

- Speak about the things they know and care about
- Listen to what others have to say
- Be counted by a system that continually aggregates and publishes the beliefs of all participants.

The required technology will extend techniques developed for BPILO and the Semantic Web. For example, when people *speak* to Consensus Builder they will not interact directly with ontological models. Instead, they will write statements and then engage in an anytime, anywhere process of interpretation that uses a variety of diagrams and other interface devices to resolve ambiguity. Speakers will be motivated to invest in this process because they want to be listened to. The greater the degree to which Consensus Builder obtains an accurate interpretation, the greater the degree to which it can reason about the statement in ways that lead to listening by other speakers, human agents active in Consensus Builder, and society in general. Figure 6 shows how Consensus Builder will communicate to speakers the quality of interpretation achieved so far, and the resulting system behaviors that are therefore enabled.

Figure 6: Feedback on interpretation of speaker statements



Model comparison will be the key to the *listening* aspect of Consensus Builder. Model comparison identifies maximal similarity and remaining differences. The best way to start a political discussion is to explicitly recognize beliefs that the participants share. Then, differences can be prioritized and considered more or less in sequence, with the goal being to encourage creative negotiation and the generation of high-quality solutions that maximize benefit for all.

Model unification will be needed to enable accurate model comparisons, especially when speaker statements are interpreted with different ontologies. To require all interpretation to use a monolithic global ontology would be unworkable. Fortunately, feedback on model comparisons can guide model unification. Thus, listening in Consensus Builder will provide a medium for building bridges on multiple levels, including communities and ontologies.

The *be counted* functionality of Consensus Builder will reflect the power of the Semantic Web, including queries that capture meaning in phrases and sentences, and responses that can be categorized and counted in ways that support insightful analysis of current belief. For example, a query might be "What kind of financial system should the United States use to pay for health care?" Statements listed in the response might be organized, for example, by stakeholders (chronically ill, health care providers, and other Americans) and preference (such as public, private, and hybrid insurance systems). Visualization and analysis of the results will increase understanding and enrich further discussion.

Trust will be the key to success for Consensus Builder. By its nature, therefore, Consensus Builder must be non-partisan, non-profit, and open source. Feedback and interest are most welcome.

References

[1] Dayne F., M. Blume, J. Byrnes, E. Chow, S. Kapadia, R. Rohwer, Z. Wang. New Experiments in Distributional Representations of Synonymy, Ann Arbor, MI. Ninth Conference on Computational Natural Language Learning (CoNLL 05), June 29-30, 2005.

[2] Elmagarmid, A., M. Rusinkiewicz, A. Sheth. *Management of Heterogeneous and Autonomous Database Systems*. 1999. San Francisco, CA. Morgan Kauffman.

[3] Parunak, H. V. "Go to the Ant": Engineering Principles from Natural Agent Systems (1/97). *Annals of Operations Research* 75 (1997) 69-101.

[4] Weinstein, P. *Integrating Ontological Metadata: algorithms that predict semantic compatibility*. 1999. Ph.D. dissertation, University of Michigan.

[5] Weinstein, P. 2006. Consensus Builder: A Place to Speak, Listen, and Be Counted. White paper. http://www.altarum.net/~pweinstein/ConsensusBuilder_draft2.pdf.

[6] Weinstein, P., Phelps, T. 2006. Business Process Interoperability with Living Ontologies. Altarum Institute Technical Report.