Business Process Interoperability with Living Ontologies

Peter Weinstein, Thomas Phelps
Altarum Institute
{peter.weinstein, tom.phelps@altarum.org}
January 5, 2006

Abstract

The Business Process Interoperability Living Ontologies (BPILO) project is developing tools for comparing models of business processes across multiple organizations. Models are represented on two levels: users work with familiar diagrams, while the system works internally with OWL. A swarming model unification algorithm converts original models expressed in heterogeneous terminology into unified models that maximize terminological and substantive commonality. Users contribute confirmation and guidance to the unification process, but are not required to provide any particular input. Unified models can be compared, yielding visualizations that crystallize insight and metrics that quantify alignment. BPILO can thus provide scientifically rigorous comparisons of business processes. Potential applications include studies focused on integration, conformance, business process reengineering, and process alignment.

1 Introduction

Business process modeling has become an important tool for civilian and defense planners as they work to improve their organizations. Unfortunately, in a cross-organizational context business process modeling often fails to deliver meaningful insights because models developed by different teams are hard to compare. Modelers use different terminology and styles, creating *arbitrary model heterogeneity* that hides genuine similarities and differences in the processes.

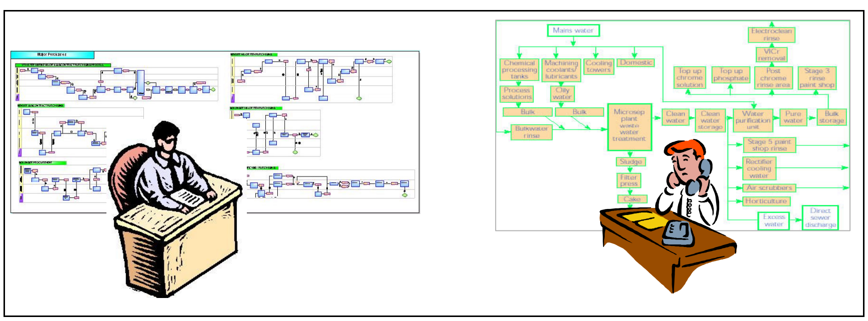

Figure 1: Business process models can be hard to compare

Arbitrary model heterogeneity occurs on three levels:

- Syntactic heterogeneity involves differences in format.
- Semantic heterogeneity involves terminology that is inconsistent (the same term is used to mean different things), redundant (multiple terms mean the same thing), or more generally, terms have meanings that overlap in ways that can be vague and/or complex. For example, two models might both say Price, but mean something different with respect to taxes, volume discounts, and so on.

- Pragmatic heterogeneity refers to differences in the application context. When there are semantic differences, the pragmatic contexts determine whether those differences are important. For example, two automobile parts may effectively substitute for each other in the context of ordinary driving, but not in the context of racing.

The Business Process Interoperability Living Ontologies (BPILO) project is developing an approach to heterogeneity that finds a productive middle ground between unrestricted use of terminology and enforced adherence to rigid standards, as illustrated in Figure 2. Unrestricted modeling of business processes yields models that are hard to compare. This is illustrated in Figure 2 by the first column of process models – one each for the Navy, Army, and a contractor – that may be essentially the same processes but because of differences between the models it is difficult to tell. At the other end of the spectrum, forcing modelers to use a limited set of terminology in a standard way – such as defined in a shared ontology – almost always yields models that are inaccurate descriptions of the processes. Thus, in Figure 2 the second column of processes are all called Purchase Requisition but forcing them into a common format that may, in fact, hide that these processes are actually not the same. The *Living Ontologies* approach, in comparison, strives to identify commonalities and tolerate differences. In the third column of the figure, each organization is expected to use the term Purchase Requisition, but they are permitted to specialize their use of the definition. In contrast to a monolithic approach to ontology, Living Ontologies yields models that are both accurate and comparable.

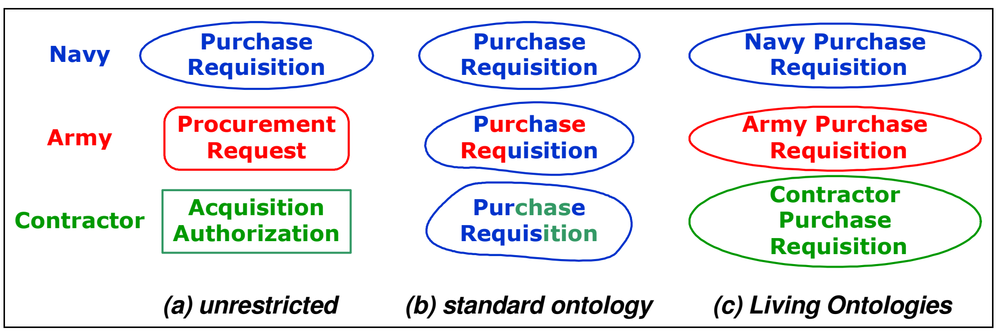

Figure 2: Three approaches to heterogeneity

This paper describes the BPILO approach to comparing models of business processes. This approach combines new technology with a user-centered stance that seeks to make reasonable demands on users as they dig into inherently difficult problems.

BPILO has three key ideas, each of which is presented in a section below. Section 2 describes the two levels of model representation in BPILO – one for users and another for computational reasoning. Section 3 describes a process of model unification that turns original models into unified models that maximize explicit commonality. Section 4 describes how BPILO compares unified models, yielding visualizations that crystallize insight and metrics that quantify alignment. Section 5 presents experimental results that describe the current state of maturity of the model unification process. Section 6 discusses potential applications and future work, and the final section provides conclusions.

2 Model Representation

A core tenet of the BPILO approach is that users should work with diagrams and other tools for thinking that are very close to the ones with which they are already familiar with through use in their daily activities. For example, business analysts use process flow diagrams, organizational charts, project timelines, maps, and so on.

As a starting point for the BPILO project, we decided to work with business process models that satisfy the requirements for the Federal Enterprise Architecture exercise OV6-c. See Figure 3. Rectangles represent processes connected by data flows, within swim lanes that identify the party responsible for each process. The figure shows a snippet from a utility that we implemented with Microsoft Visio. The learning curve for this utility is almost zero for users that already know Visio and business process models. A custom Visio stencil provides shapes for processes, flows, and swim lanes that modelers drag onto the page, link and label.

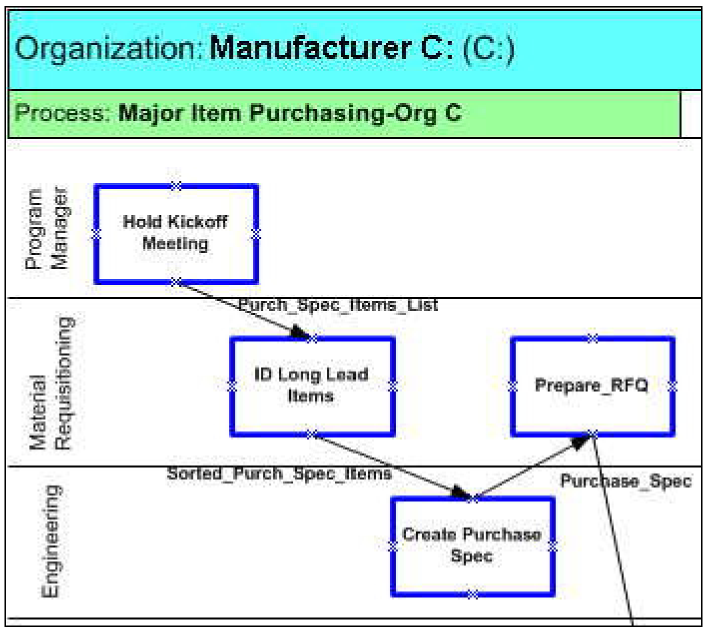

Figure 3: Fragment of a business process model

Internally, the BPILO system represents these process models with description logic, namely the Web Ontology Language (OWL). The formal ontological representation provides a basis for the computational reasoning described in the following sections, as well as the standard forms of reasoning supported by description logic including automatic classification using subsumption inference. Currently, we use a subset of OWL-DL that prefers existential rather than universal quantification and avoids negation and disjunction.

We do not ask users to specify formal ontological models directly because we find that people who lack training for this kind of work usually flounder. We find that the activity of specifying formal ontological models is very much like designing a class structure for object-oriented software systems. Few managers would require people without training in object-oriented design to undertake such an activity. Instead, BPILO users click buttons to export models from Visio into XML, then import into the BPILO system.

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 user diagrams into ontological models turns meaning *implicit* in the diagrams into *explicit* relations amenable to computational reasoning. For example, the location of a process shape within a swim lane in the Visio diagram means that the responsible party for that process is identified by the label of the swim lane.

Figure 4 shows the definition of ID Long Lead Items, as specified in Figure 3, after importing into the BPILO system. The BPILO system is implemented using Java as an extension of Protégé, the ontology editing tool being developed at Stanford [Knublauch et al. 2004]. The upper part of the definition has been instantiated from the business process model. The lower part is inherited from the core definition of Process. For example, this concept does not have further detailing into subprocesses, so the open world existential property of hasSubTask is filled with the generic expectation of Task.

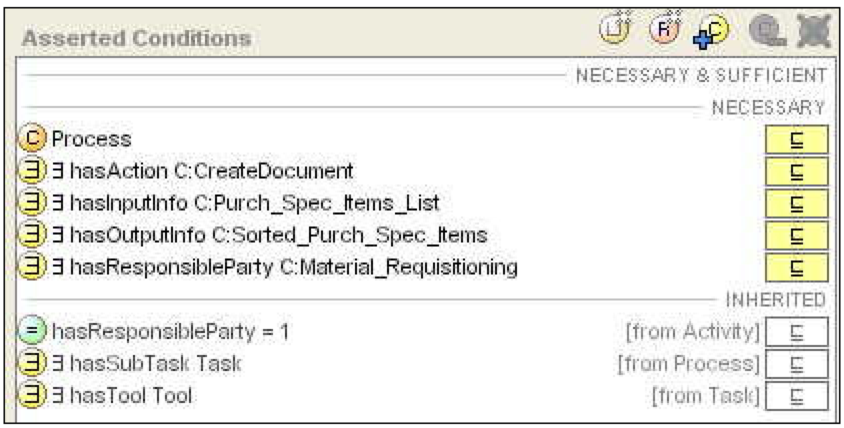

Figure 4: Definition of the concept C:Select_Vendor_Orig in Protégé

3 Model Unification

Model unification converts heterogeneous process models produced by multiple organizations into integrated models that can support accurate comparisons. This section describes the swarm intelligence algorithm that we use to achieve model unification, and the *anytime, anywhere* user interaction made possible by this technique.

3.1 Swarming 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 5, where heterogeneous organization-specific concepts inherit some properties from core concepts that are shared (blue layers are shared, the red is not).

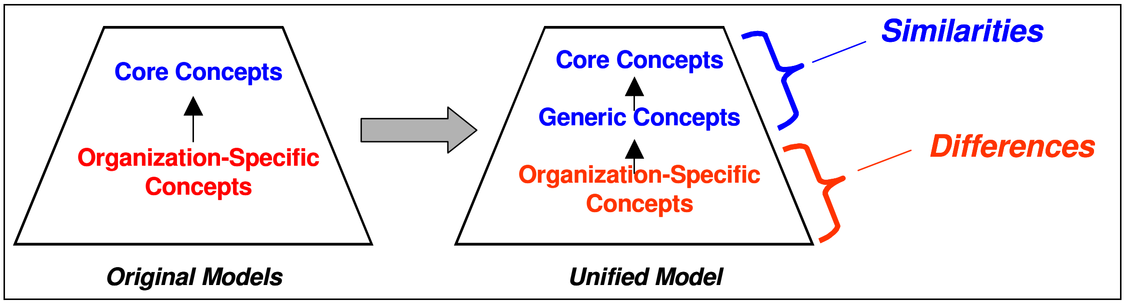

Figure 5: Before and after model unification

Model unification converts original models to *unified models* by creating a shared middle layer of generic concepts from which organization-specific concepts inherit. The goal of the unification process is to make the definitions of the middle layer of generic concepts as substantial as possible – thus maximizing the degree of explicitly identified commonality. In the unified models, properties associated with organization-specific concepts represent differences, while properties inherited from the generic layer represent similarities. Therefore, the fundamental ability of ontologies to organize concepts with inheritance makes it possible to make similarities and differences explicit in the unified models.

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 [Parunak 1997]. A swarming architecture can provide an algorithm with several benefits, including:

- Efficient identification of near-optimal solutions to hard problems
- Highly parallel execution yielding scalability to handle massive data
- Graceful adaptation to runtime change in the problem (picture ants adapting when, for example, a food source is moved to a new location).

The last feature is particularly valuable for creating anytime, anywhere user interaction, since each user input essentially changes the problem that the swarming algorithm is in the middle of solving.

Our strategy for model unification is to design fine-grained agents that play roles defined by their places in the unified models. For example, the role of an agent associated with generic concepts in the unified models is to find the organization-specific concepts that are best matched so as to inherit as much as possible from the generic concept. Meanwhile, the role of agents associated with the organization-specific concepts is to find a match where they best belong. The swarming unification process thus has agents of various types that make decisions trying to maximize local criteria. The resulting unified model emerges from this activity.

The swarming unification algorithm has three concurrent processes:

- <u>Generalization</u> matches corresponding elements of the original models to create generic concepts
- <u>Granularity adjustments</u> split and join elements to increase alignment
- <u>Rewriting</u> of organization-specific concept labels to highlight commonalities.

Of the three processes, generalization is the most central. This process is currently implemented and is the focus of the remainder of this section and Section 5.

We describe the generalization algorithm by analogy to the game of musical chairs. This is because only one concept from each original model is allowed into a match (this is a reasonable assumption because, by design, the granularity adjustments process will create concepts that can be matched on the same level of detail). For example, in Figure 6 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 displacing the C:Create_PO concept from that match.

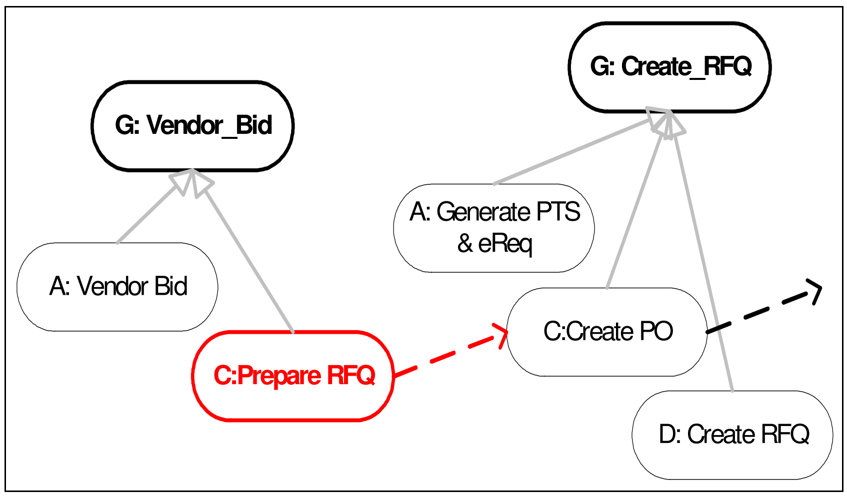

Figure 6: Concept matching as musical chairs (a concept joining a match kicks another out)

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 [Dayne et al. 2005]. These estimates are based on word co-occurrence in a corpus of documents describing business processes. We had trouble finding an appropriate corpus of documents and so assembled one ourselves, containing over 700 documents and about 31 megabytes of text.

We also estimate structural compatibility in the sense of isomorphism. In other words, 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. For example, in Figure 7 the concept C:Prepare RFQ is matched to D:Create RFQ. If C:Purchase Spec is also matched to D:Request, then this latter match causes a higher similarity estimate for the former match.

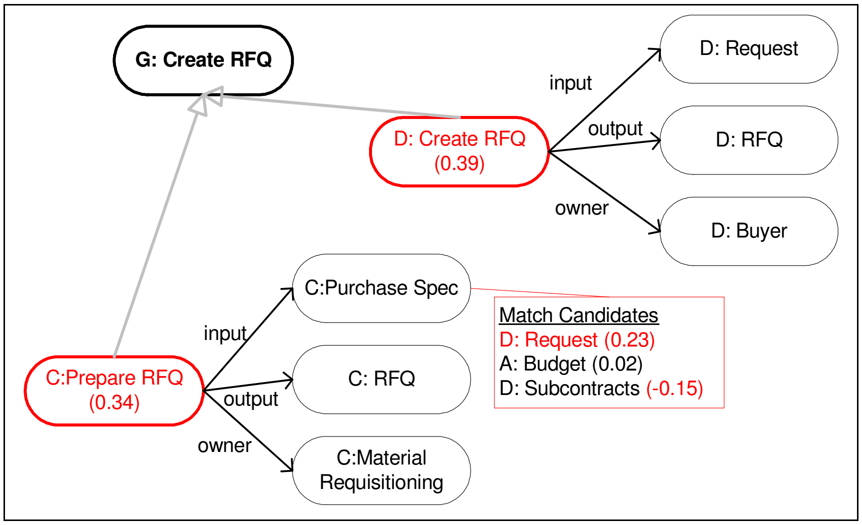

Figure 7: Matches suggest further matching of structural relatives

As the unification process progresses we shift attention from lexical association to structural similarity. Initially lexical similarity is weighted higher, but structural similarity is more important at the end. Thus the words provide clues that lead us to structural isomorphism. For example, each of the original models in our test problem includes a concept called Requisition. In a classic manifestation of semantic heterogeneity, however, this term means something different in each of the organizations. At the beginning of model unification these concepts are often matched together, but by the end of the process they are separated.

Structural relations are also used to accumulate suggestions to agents associated with original concepts regarding which matches to join. For example, the match between C:Prepare RFQ and D:Create RFQ in Figure 7 causes the match agent to suggest to the C:Purchase Spec and D:Request agents that they should consider joining together in a match. These suggestions are called *pheromones* in the lingo of swarming, and like the chemicals that ants deposit they also evaporate over time. The higher the quality of the match, the more pheromones are deposited. By analogy, if I marry a woman and am happy, my brother may spend some time with her sister and also decide to get married. Model unification thus has a positive feedback effect, as is typical of swarm intelligence algorithms, since structural isomorphism encourages further matches that express that isomorphism.

Another typical feature of swarming algorithms is to reduce the degree of randomness in decision-making as a process moves forward. This is called cooling the temperature since the idea is taken from simulated annealing. The advantage of this technique is to expose agents to many possibilities and avoid local minima in the optimization landscape. Both the simulated annealing and the positive feedback of structural isomorphism lead us to expect a crystallization effect in model unification, where the slope of the total similarity found suddenly increases and then levels off.

3.2 Anytime, Anywhere User Interaction

There are two principles for user interaction that we are seeking to achieve in BPILO:

1. Interact using the user's representations not the system's. For example, interaction should focus on process diagrams not on ontological models.

2. Let the user confirm and guide processing without requiring any particular input from them at a rigid time (interaction should be anytime and anywhere).

Anytime, anywhere interaction is vital for model unification because at the point where it is necessary to learn the processes of other organizations, this becomes a hard problem. If in typical computer style the system demands that users deliver the key insights so that it can proceed, most users will give up right then. In comparison, anytime anywhere interaction should be relatively pleasant – or, at its best, even have a playful quality that makes the work fun.

To date, we are making good progress on these goals for user interaction during model unification, but we are still short of where we want to be. This section presents the user interface in the current BPILO system, and describes the interface that we are planning to develop.

Figure 8 shows the window that provides an overview of the generalization process. Note the Process Control Toolbar, which has two parts. On the left, VCR-style controls cause the process to play, pause, and stop. On the right, various metrics provide a snapshot summary of the state of the process. In the main body of the window, the user selects a perspective (usually his or her organization but there is also a generic view), and a subset of type of concept (e.g., Process, Data, Flow, or Responsible Party). The two lists in the window actually include the same list of concepts, but sorted such that concepts at the top of the left list are those in strong matches (with a high similarity estimate) while the concepts at the top on the right are those in weak matches. We choose to present the list twice because the appropriate user action is different in each situation: with respect to good matches the system is looking for confirmations, while for poor matches the system is looking for more active guidance. Selecting a concept and clicking a Provide Feedback button causes a drilldown to the next window. Generally, users must pause the process before taking an action because the display is constantly changing otherwise. In situations where it is possible to corrupt the state of the system the process is automatically paused when a user begins an action.

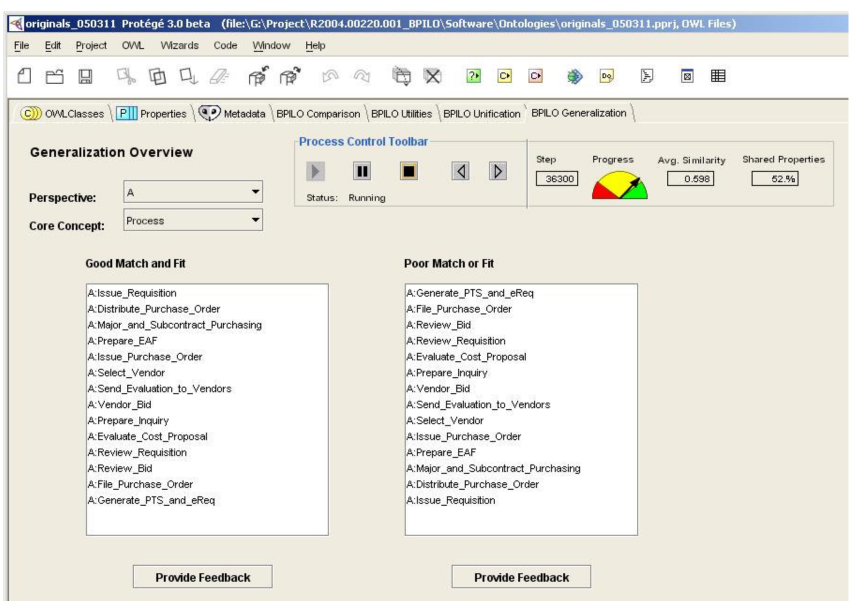

Figure 8: Lists of good and poor matches

Figure 9 shows the currently implemented view of match formation. The column on the left describes the current match: its name, the properties that all of the matched concepts have in common, and the names of the concepts that are matched. The name of the match is selected from the matched original concepts, using the name that has the highest average lexical association with the other concept names.

The middle column focuses on differences: the properties that are not shared in the focus concept (which was selected in Figure 8), and of the match member currently selected in the list to the left.

The right column shows accumulated suggestions for other matches that the focus concept should consider joining (sorted by the accumulated strength of the pheromones), and the concepts that are currently in those matches.

Users can take several kinds of action. They can confirm that one or more concepts belong in the current match. This can be done in either a tentative or sure manner. A tentative confirmation strengthens the suggestion to be in the current match and removes suggestions to other matches. Since these pheromones evaporate over time and new suggestions are constantly generated, however, a tentative confirmation may not have a lasting effect. A sure confirmation pins concepts to a match for the duration of the process. Users can also remove concepts from the match – they are temporarily kicked out to singleton matches but from there will quickly find a new home. It will also be possible to add a concept to the current match. Finally, users can move the focus concept into another match identified by its current suggestions.

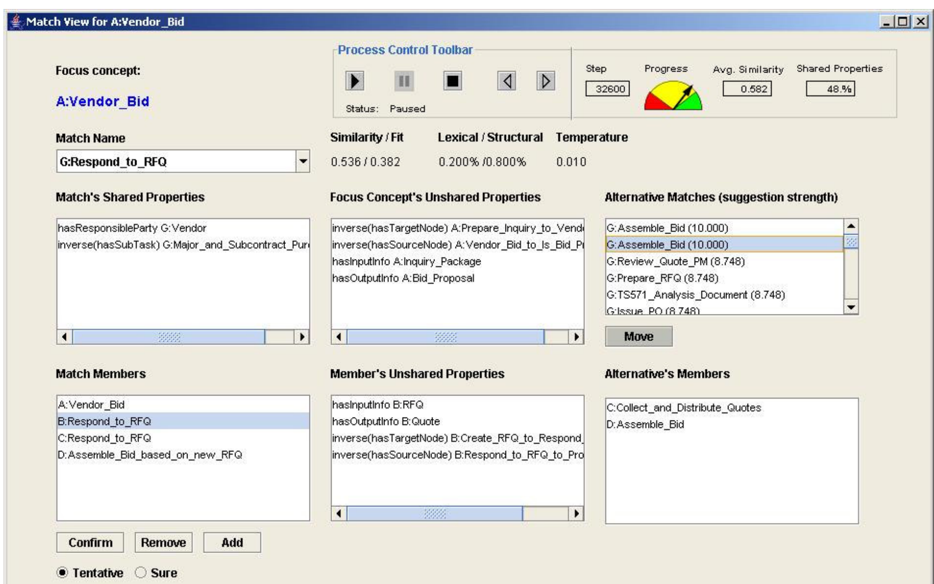

Figure 9: Detailed view of match formation

While the current implementation of the BPILO model unification interface is anytime and anywhere, the presentation has little visual impact and falls short of the aspiration for a playful user experience. Figure 10 illustrates a window that would be informationally equivalent to Figure 9 but more engaging. For example, the "musical chairs" analogy could be conveyed with small animations that show one concept kicking another out of the match.

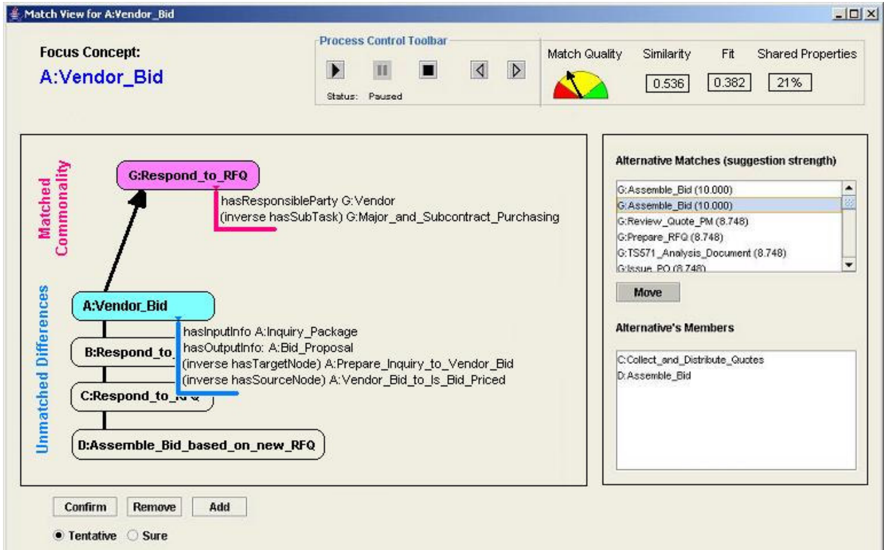

Figure 10: Mockup of graphical view of match formation

Unfortunately, all of the user interface windows illustrated in this section are focused on the wrong level of detail – on the ontological model rather than process flow diagrams. There is a big gap between these two levels of representation and some calculation needed to bridge the gap, for example to draw a process flow from its ontological model.

The user interaction that we are planning to develop will show pieces of the emerging generic process model as they clarify and link. The user will be able to watch this self-organization and participate in it. To make this work will first require implementation of the granularity adjustments and rewriting processes for model unification. There will also be heterogeneity issues involving the flows that link processes, because there tends to be a high degree of variability in the order of process steps across organizations. Finally, we anticipate a need for some fancy user interface programming.

We call our approach to knowledge representation Living Ontologies because the population of concepts is constantly changing. Users will introduce new organization-specific concepts, and new shared concepts will be created during various unification processes where selected source models are integrated to support various kinds of comparative studies. Likewise, many concepts will become obsolete as the problems they help to address evolve. Thus, whereas ontologies are often thought of as a form of computable dictionary, Living Ontologies are more like an evolving compilation of project glossaries.

4 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 [Weinstein 1999]. Figure 11 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 needs to be viewed in color). Elements of the concept definitions that are modeled by the generic layer are shown in pink. The pink therefore conveys similarity, while blue and green are differences. For example, the box labeled C:Meet_Qualified_Vendors is green because organization C conducts this process step while organization A does not. Many boxes are tricolor because the some but not all of the properties of the original concepts are matched. Thus, users will be able to click on elements of the visualization to drill down into the model and see in detail what is shared and what is different.

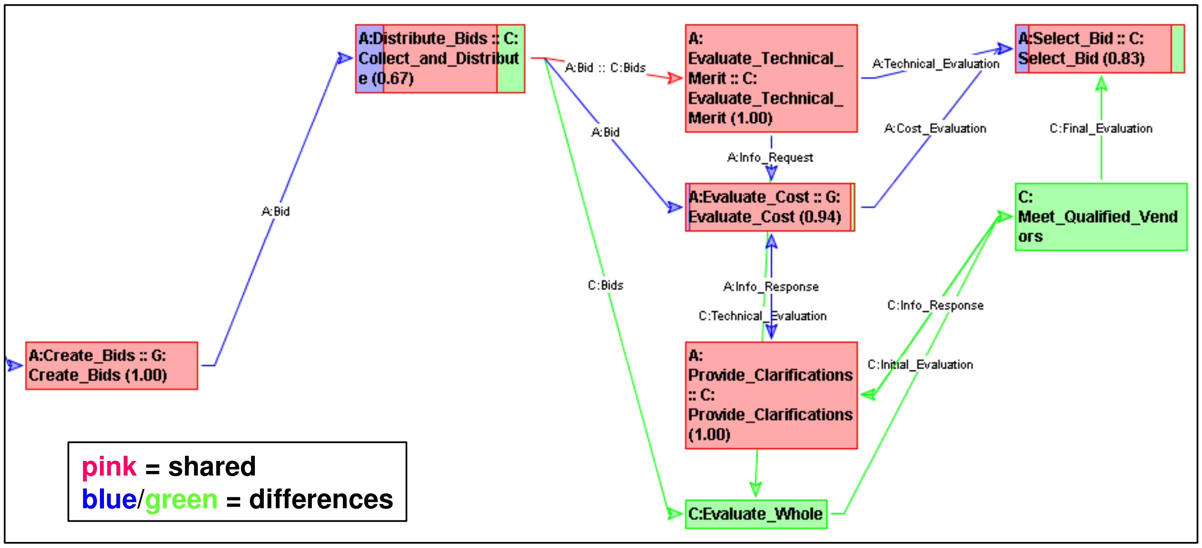

Figure 11: Visualization of a process comparison

Furthermore, each match and the overall comparison are quantified with a metric that describes the degree of commonality, based on a weighted combination of lexical association and structural isomorphism, yielding a number in the interval [0, 1]. This quantification creates potential for adding a new level of scientific rigor to process comparisons. The model comparison algorithm that we have currently implemented, however, all properties are treated as equally important and the metric boils down to how many are shared and not shared.

In any particular pragmatic context, however, some differences matter a lot while others can be safely ignored. There is therefore a need for models of what is important for particular applications. These models will be utilized by the comparison algorithms to generate similarity metrics that are accurate and practical for particular purposes.

The models of pragmatic context need not necessarily be ontological in nature: for example, some situations call for representations such as Bayesian networks that model the probabilities of various kinds of interactions. In previous research on semantic compatibility we developed families of algorithms for quantifying similarity that have various properties [Weinstein 1999].

5 Unification Experiments

This section reports on experiments on the model unification process. The subsections describe the test problem, experimental methodology, and the results.

5.1 Test Problem

To achieve operational validity, for original models we used real as-is models of purchasing processes developed by one of the authors (Phelps) in an engagement with four medium sized manufacturers. (We need to be vague about the identity of the manufacturers to protect corporate privacy). The models were initially captured using a variant of value stream mapping. They include a total of about 250 concepts, including 61 organization-specific processes, 57 data items, 71 flows, and 29 responsible parties, with a balance of other miscellaneous types of concepts.

Several factors of the capture process may have led to a greater degree of homogeneity across the models than might sometimes be the case:

- The basic characteristics of the companies are similar, e.g. they are of similar size and in the same industry
- The same person authored each of the models
- The modeling process was not affected by negotiations or other kinds of real-world politics that often lead to models that are less clear than one might wish.

By design, however, the input process models used terminology as it was used by the subject matter experts interviewed in each organization. The models were originally developed with heterogeneous terminology because their original purpose was for communicating with the members of each organization, separately. Also, when the current BPILO tool is used to compare these as-captured processes, it finds very little commonality, thus demonstrating a high degree of heterogeneity.

After identifying the original models, our first step was to manually define a unified model. This unified model served two purposes. First, it let us illustrate the benefits of having a unified model before we were able to generate it automatically. Second, it provided a standard against which we could measure unified models generated automatically or semi-automatically by the system.

The process of manually defining a unified model took the authors approximately one day. When we examined the purchasing processes using our specific knowledge of those companies, we were able to discern a common three-step top-level purchasing process including Specify Purchase, Select Vendor, and Complete Purchase. We then articulated the second and third of the top-level steps (the first step had been out of scope for the original data gathering) into more detailed generic processes. To accomplish this we drew boundaries between the three top-level steps in each of the original models, then tried to match the concepts within each section across organizations. The resulting unified model includes 57 concepts defined at the generic level.

We do not see our manually unified model as an ideal model or the best that could possibly be defined. A lot of intelligence and knowledge went into its specification, however, so we believe it is an appropriate baseline for comparison.

5.2 Methodology

The goal of these experiments was to compare the unified models generated by the BPILO system against the manually defined unified model.

We ran six experiments, each including a sample of ten runs. The experiments vary in their simulated level of user contribution. The first experiment had no user contributions. In each experiment, an increasing portion of concepts is glued into their matches as if the user had used the “sure confirmation” button to do this. In the final experiment, half of the original concepts are glued. We did not simulate actions other than confirmation, such as for example users moving concepts into their correct match.

The algorithm that simulates the confirmations starts with the match with the highest estimated similarity (which will show at the top of the Good Matches list in the user interface). It looks for two concepts that are matched in the manual model, where at least one of the concepts has not yet been glued into place. If such a pair is found it is glued into place; otherwise, the next best match is inspected. When two concepts are glued into place, the next user confirmation is skipped so the final number of confirmed concepts is almost exactly as targeted for the experiment.

Each run included 30,000 processing steps, where each step provides a chance for some concept agent to move to a different match. The runs executed in approximately 0.5 minutes on a 1.7Ghrtz laptop. Clearly, realistic use scenarios model unification would be allowed to take much longer. We found that longer runs did not significantly improve the results, however, probably due to the small size of the test problem.

At the beginning of each run, similarity was calculated as 80% lexical association and 20% structural isomorphism. This ratio changed in a linear manner until by the final step the ratio was reversed. The temperature of each run was also dropped linearly from 0.5 to 0.1. To give an intuitive understanding of these temperatures, let us say that an agent must decide between two options, one of which is rated at 0.6 and the other at 0.4 where a higher rating is better. At the beginning of the runs, the agent would have a probability of choosing the first option that was a few points greater than 50 percent. Near the end of the run, the agent would be almost sure to choose the first option.

The metrics that we use to compare the generated unified models against the manual unified model are designed to answer the following questions in as straightforward a manner as possible:

- To what extent do we succeed in generating the matches in the manual model?
- To what extent do the matches that we generate approximate those in the manual model?

The difference between these questions is one of perspective: in the first we start with the manual matches and ask how close are the generated matches, while in the second we start with the generated matches and ask how close these are to the manual matches.

We also decided to focus our attention exclusively on the quality of the matches that generate generic Process concepts, as opposed to Data, Flows, and so on. The rationale for this decision is primarily that our development of the test problem focused on processes. These concepts essentially lie in the middle of small webs of connections. The other types of concepts are somewhat ancillary and this incompleteness could strongly impact the unification processes. In particular, the test problem does not model the structure of the data.

There are 61 original concept agents in our experiments trying to find matches where they best belong. 40 of these are matched in the manually unified model. The manual unified model contains ten generic Process concepts with multiple organization-specific subconcepts. The manual model includes aggregation of concepts, however. In our metrics, we drop extra concepts from an organization in the same match. Thus, many of the manually unified matches contain fewer than four original concepts. We also dropped two manual matches where a concept modeled as a decision in an original model was modeled as a process in the unified. This left eight remaining manual process matches that are the focus of the results presented in the following section.

5.3 Results

The results in the first part of this section show that the generalization process is behaving as designed to maximize commonality in the unified model. The second subsection compares the generated models to the manually created unified model. The results from this perspective are not as clear.

5.3.1 <u>Generalization at Work</u>

The generalization process is designed to find matches that maximize the similarity of the concepts that are matched. Figure 12 shows the average pairwise similarity of matched concepts as runs progress. Similarity as shown is a weighted combination of lexical and structural similarity measures – 80 percent structural and 20 lexical. This is the similarity weighting used by the end of the process, having started at a weighting of 80 percent lexical and 20 percent structural. In this and following graphs, results are averaged across the runs for each experiment. Figure 12 shows three data series: for the experiment with no simulated user confirmations, for the experiment where 30% of concepts are confirmed, and a static baseline that shows average similarity of matched concepts in the manual unified model.

Several features of Figure 12 are notable. First, the curve of rising match quality has the expected S shape, due to the falling temperature and the crystallizing effect of positive feedback on structural isomorphism. Second, the final quality achieved is substantially greater than the unified model. We will discuss the implications of this behavior below. Third, the inclusion of user confirmations of close to one third of all concepts has a surprisingly small effect on the similarity of the resulting matches.

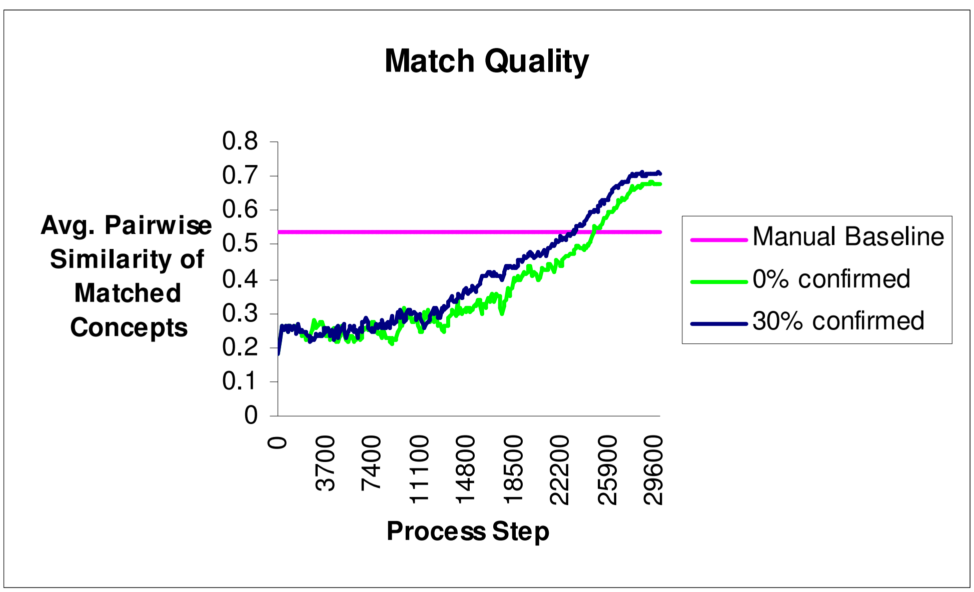

Figure 12: Creating matches of similar concepts

Figure 13 shows the average portion of properties of the original concepts that are inherited from the generic concepts generated by their matches produced by unification. In other words, in the generated unified models significantly more than half of the definitional substance of the process concepts is shared. As in Figure 12, however, the degree of commonality generated is substantially greater than that found in the manual unified model, and there is little difference between runs with zero and 30% confirmations.

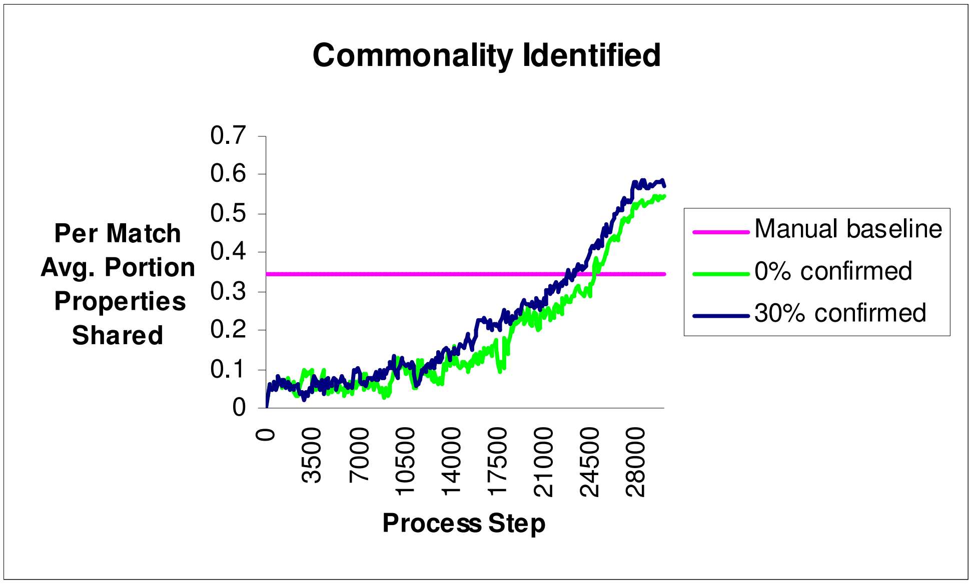

Figure 13: Maximizing the definitional substance of the generic layer of concepts

5.3.2 Recreating the Manually Unified Model

This section compares the results of the generalization processes against the manually unified model. We developed two metrics that look at the question from each perspective:

- Manual Match Satisfaction (MMS) – To what degree are the desired manual matches created by automatic generalization?
- Generated Match Desirability (GMD) – To what degree do the generated matches contain concepts that should be together according to the desired manual matches?

Each manual match can be thought of having four slots, one for each organization. Each slot can contain one or zero concepts. The MS metric is based on the generated matches that come closest to filling its slots as does the manual match. The score for each manual match is one of {0.25, 0.5, 0.75, 1.0}, depending on how many slots are filled as desired in the generated match that best satisfies the manual match. Thus:

(Eq 1) $$MMS = \sum_{m \in MM} \sum_{i=1 to k} \frac{\delta(S_i(m) = S_i(B(m)))}{k} / |MM|$$

where MM is the set of manual matches, S is a slot function that returns the concept for organization i in match m (or empty), k is the number of slots, B is a function that finds the generated match that best satisfies manual match m, and δ is an indicator function δ→{0,1} that reflects whether the manual match and the generated match fill the slot with the same concept or whether both are empty.

The Generated Match Desirability (GMD) metric shows the average percentage of pairs of concepts in generated matches that are also paired in the manual matches:

(Eq 2) $$GMD = \sum_{m \in GM} \sum_{i,j \in c, i \neq j} \frac{\delta(c_i, c_j)}{|c|(|c|-1)} / |GM|$$

where GM is the set of generated matches, and δ is an indicator function δ→{0,1} that reflects whether the generated match between concepts i and j in the set of concepts c defined by match m are also matched in the manually unified model.

Figure 14 shows the Manual Match Satisfaction metric over the course of generalization processes with zero and 30 percent user confirmations. We would characterize the curves in Figure 14 as positive but weak.

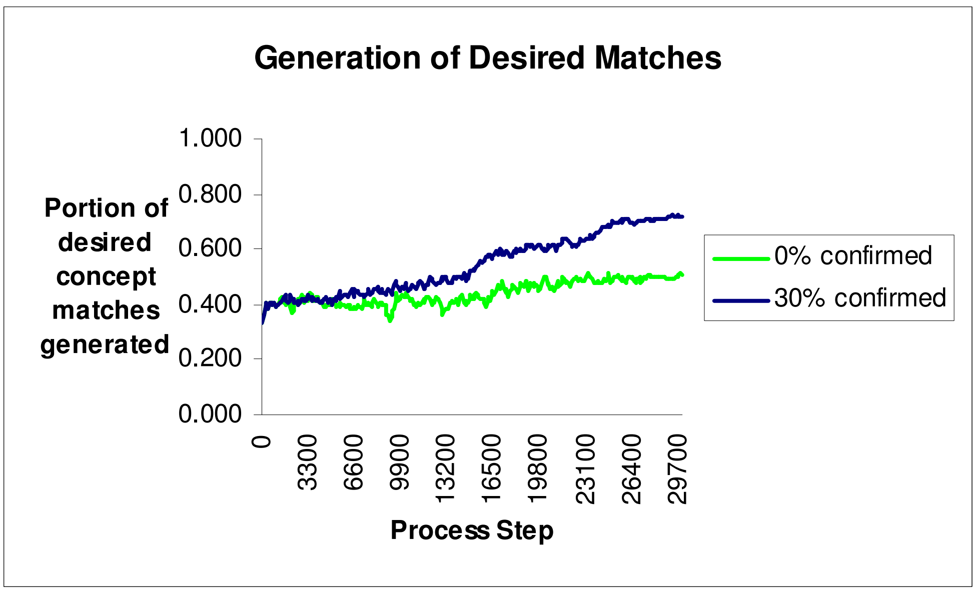

Figure 14: The degree to which desired matches are generated

Table 1 lists each desired manual match and the contents of the generated match that best satisfies it that was produced by a particular run with 30 percent user confirmations. In this run, half of the desired matches are recreated perfectly. Two of the weakest matches were cases where the manually unified model aggregated two or more of the original concepts, making it unlikely that automatic unification would recreate the match until the granularity adjustments process is implemented.

Table 1: Generated matches that best satisfy the manual matches (at 30% user confirmations)

Manual Match (MMS)	Desired Original Concept	Generated Match Concept (blue – on target, red – not)
G:Create_Bids (1.0)	A:Vendor_Bid B:Respond_to_RFQ C:Respond_to_RFQ D:Assemble_Bid	A:Vendor_Bid B:Respond_to_RFQ C:Respond_to_RFQ D:Assemble_Bid
G:Solicit_Bids (1.0)	A:Review_Requisition B:Create_RFQ C:Prepare_RFQ D:Create_RFQ	A:Review_Requisition B:Create_RFQ C:Prepare_RFQ D:Create_RFQ
G:Purchase_Item (1.0)	A:Major_and_Subcontract_ Purchasing B:Major_Item_Purchasing C:Major_Item_Purchasing-Org_C D:Purchasing-Org_D	A:Major_and_Subcontract_ Purchasing B:Major_Item_Purchasing C:Major_Item_Purchasing-Org_C D:Purchasing-Org_D
G:Create_PO (1.0)	A:Prepare_EAF B:Create_PO C:Create_PO D:Create_PO	A:Prepare_EAF B:Create_PO C:Create_PO D:Create_PO
G:Evaluate_ Technical_Merit (0.75)	A:Review_Bid B:Evaluate_Quotes C:Review_Quote_Tech < empty >	A:Review_Bid B:Evaluate_Quotes C:Review_Quote_Tech D:Assemble_Bid_based_on_new_RFQ
G:Select_Bid (0.5)	A:Select_Vendor B:Clarify_Issues_with_ Vendor_and_Select C:Select_Vendor_Orig < empty >	A:Send_Evaluation_to_Vendors B:Clarify_Issues_with_ Vendor_and_Select C:Review_Quote_PM < empty >
G:Distribute_PO (0.5) * includes aggregation	A:Issue_Purchase_Order B:File_Supporting_Docs C:Issue_PO D:Fax_PO_to_Vendor	A:File_Purchase_Order B:File_Supporting_Docs C:Issue_PO D:Issue_PO
G:Evaluate_Cost (0.5) * includes aggregation	A:Evaluate_Cost_Proposal B:Process_Quotes < empty > < empty >	A:Evaluate_Cost_Proposal B:Process_Quotes C:Collect_and_Distribute_ Quotes D:Finalize_RFQ

Figure 15 shows the Generated Match Desirability metric over the course of generalization processes with zero and 30% user confirmations. The improvement over the course of the runs seems more impressive from the perspective of the generated matches than from that of Figure 14, especially in the run including user confirmations. It is interesting that the average improvement levels out near the end of the runs. We note that this dovetails with the points in Figure 12 and Figure 13 where the unification metrics surpass the unified model.

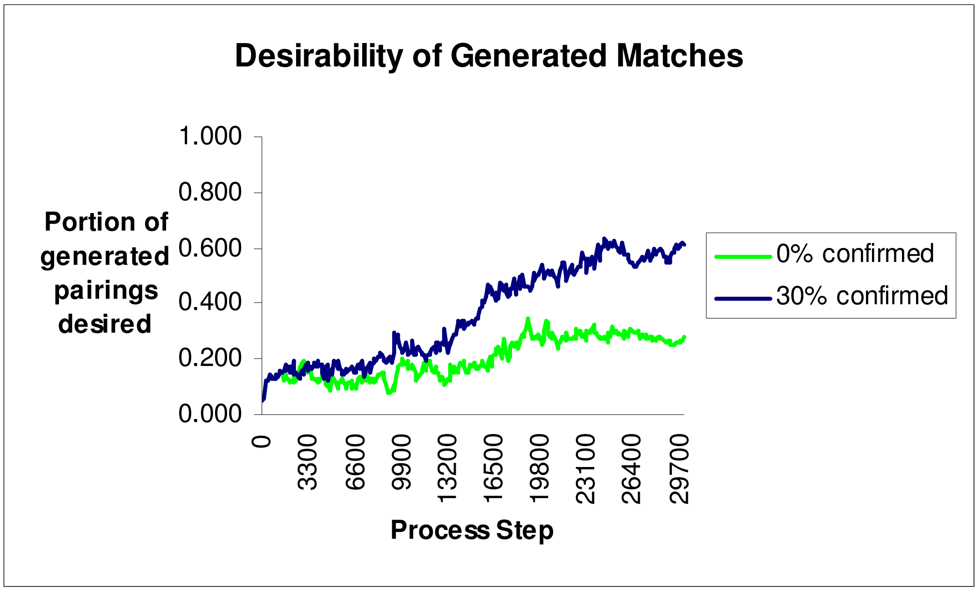

Figure 15: The degree to which generated matches have desired pairings

Figure 16 and Figure 17 show our metrics for comparing the manual and generated runs across all six experiments. Unlike the graphs above, here we are looking only at final results, which is what matters the most. As one would expect, user confirmations do help substantially. There is a hint of an inflection point somewhere around a rate of 30% confirmations, especially in Figure 17. This suggests that a certain number of user confirmations may help the system automatically find additional desired matches.

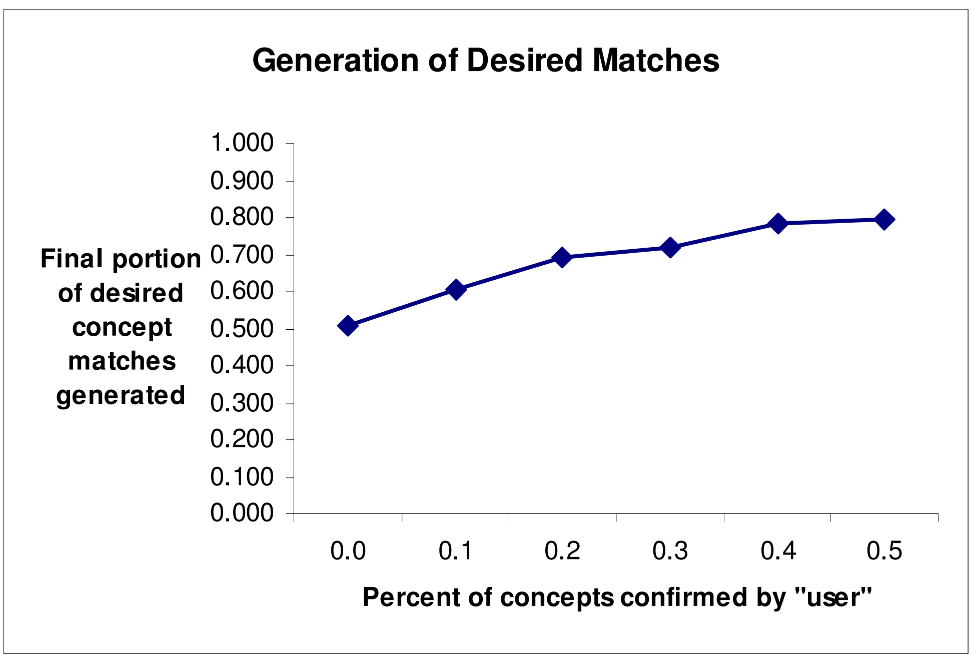

Figure 16: Helpfulness of user confirmations for generating the desired matches

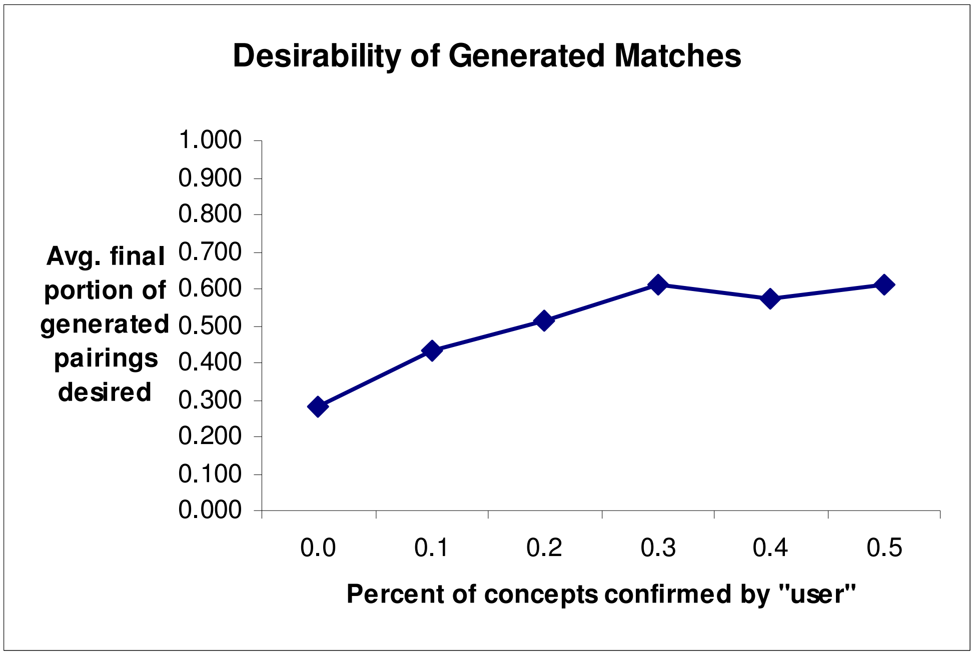

Figure 17: Helpfulness of user confirmations for generating desired matches

Finally, Figure 18 describes the ease with which the BPILO system can tell which of the generated matches should be turned into generic concepts in the unified model. More precisely, when the generated matches are sorted for average pairwise similarity, how many of the matches that best satisfy one of the eight desired matches are ranked in the top eight of the generated matches? Typically there are between twelve and sixteen generated matches. Therefore, the degree of selectability is somewhat but not substantially higher than what would expect by merely random selection. And, selectability does not increase with the quantity of user confirmations. If a clear threshold for generating generic concepts from matches is not available, then our default strategy will be to generate concepts from all matches.

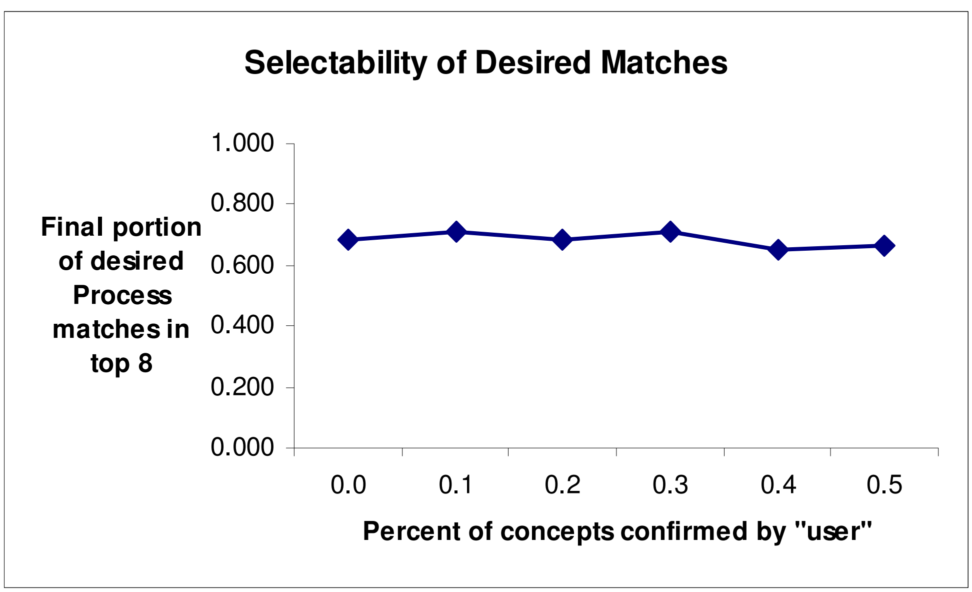

Figure 18: Degree to which the system can identify which matches are desired

Overall, we consider these results to be promising but inconclusive. The manual unified model is less strong than we would like from the point of view of the metrics that the system is driving to optimize: the similarity of matched concepts, and the amount of commonality captured by the unified model. During the process of specifying the manual unified model, Phelps in particular applied deep knowledge deriving from his personal visits to each of the manufacturers. Meanwhile, only a tiny part of that knowledge was represented in the original models. We thus believe that our results will become significantly stronger when we expand the scope of knowledge captured in the original models. In particular, we believe it will be helpful to model the structure of the data passed between processes, and to model multiple levels of processing detail. We also suspect that this further elaboration of the input models will be necessary to develop the granularity adjustments process of model unification, which will involve aggregating and occasionally splitting process concepts.

6 Discussion

This section discusses potential applications for the BPILO technology, reviews related work, and identifies priorities for continuing our research.

6.1 Potential Applications

Potential applications of the BPILO technology boils down to that models that are being compared. Typically, people talk about AS-IS models, which describe business processes as they are currently practiced, and TO-BE models, which describe processes as they will be practiced after a successful transformation. Standards describe processes as they should be practiced. There is also a grey area between AS-IS and TO-BE models, as planners try to figure out how to get from the AS-IS state to the TO-BE goal. Thus, some planners speak informally about "AS-BE" models.

Table 2 provides a summary of the types of projects defined by the types of models to be compared. The objective of interoperability underlies all of these types of projects, although perhaps this is not necessarily true for business process reengineering where the goal may possibly be to achieve improvements solely within the organization. When multiple organizations are seeking to transform themselves simultaneously, such as within the U. S. federal government today, alignment studies become particularly important.

Table 2: Potential applications for BPILO

Type of Project	Model 1	Model 2
Integration	AS-IS	AS-IS
Conformance	AS-IS	Standard
Business Process Reengineering	AS-IS	TO-BE
Alignment	TO-BE	TO-BE

Even more generally, we see potential for using Living Ontologies to create knowledge sharing and improve effectiveness in almost every area of our society. For example, we have already considered potential applications for unifying parts catalogs, integrating intelligence data, comparing medical practices, and supporting political discourse.

6.2 Relevant Research

The potential for improving intra- and inter-organizational business processes has been a central goal of the Semantic Web movement [W3C 2001]: for example, see [Daconta et al. 2003]. Most attention has focused on using the Semantic Web as a mechanism for achieving interoperability. For example, see [IBM 2005].

From the technical perspective, model unification can be seen as an application of ontology alignment. The problem of ontology alignment takes as input two ontologies, and produces as output a set of equivalence and subsumption relationships between concepts in the different ontologies.

In the last couple of years, research on ontology alignment has reached a new level of maturity. For example, there are now annual contests where researchers can pit their techniques against each other using a common set of problems and evaluation metrics [OAEI 2005]. See [K-CAP 2005] for a downloadable set of publications from a workshop discussing experiences in ontology alignment.

Most ontology alignment systems use of blend of techniques to estimate similarity between concepts in a manner much like BPILO's use of lexical association and structural isomorphism. Other sources of semantic information are also used to good effect such as string comparison of concept names, broad coverage ontologies such as WordNet, and natural language processing. The implication of this research for BPILO is that the accuracy of the automatic component of model unification can be improved by incorporating additional sources of semantic insight. We suspect, however, that there will be a diminishing marginal benefit for such improvements since one would expect there to be strong correlation between the various sources of semantic information.

Frequently, systems for ontology alignment do involve users since the problem clearly requires deep understanding to get the job done well. We are not currently aware of other research, however, in which user interaction is anytime, anywhere, or in which the goal is to focus interaction on a level of detail that is much more accessible to organization managers rather than expecting them to engage directly in the details of ontological modeling.

The organizational context that has most directly influenced the development of BPILO has been the effort by the U. S. federal government to transform its agencies to comply with a Federal Enterprise Architecture [FEA 2005], an articulated by a set of prescriptive reference models for data, processes, and so on. The Defense Department has developed an initiative [BMMP 2005] that is now publishing a substantial set of architectural guidelines available on the web after a registration that for us was approved within the same day (look for the BMMP Business Enterprise Architecture). Because it is impossible to enforce strict compliance to architectural standards that have both substance and precision, the evolution of the enterprise architecture movement seems to be moving in the direction of federated architectures: namely, that standardization can occur on a relatively high level of abstraction, while permitting the details of processing to differ from organization to organization.

The federated architectures idea is harmonious with the Living Ontologies approach, which seeks to identify commonality and tolerate differences among processes. In the enterprise architecture movement, the will to transform in the enterprise architecture movement is flowing from the top down. The BPILO technology can be used to measure conformance in a top down effort, but it can also enable bottom up movement towards the same objective. If Living Ontologies technology can be used to more fully engage managers at relatively lower levels of government organization, then it may be possible to achieve a more thorough realization of the desired transformation.

6.3 Work to be Done

We identify the following priorities to push the BPILO technology forward in the near term:

- Develop model unification problems that include models of data and multiple levels of process detail
- Implement the granularity adjustments process to execute concurrently with generalization during model unification
- Design and implement anytime, anywhere user interaction with model unification using business process diagrams, rather than focusing on match definition which corresponds directly with the relatively detailed internal representation in OWL.

In the bigger picture, our challenge is to learn how to embed the high tech BPILO system within high touch change management practices that help organizations transform to meet their long-term goals. Many factors must be considered to achieve effective transformations including organizational cultures, personal politics, and so on. BPILO may create a potential for achieving transformations with increased scientific rigor, but can by no means ever be expected to guarantee success.

7 Conclusions

The Business Process Interoperability Living Ontologies (BPILO) project is developing tools for comparing models of business processes. The approach is practical and oriented towards building tools that help consultants work with organizations to transform their processes to increase interoperability, flexibility and effectiveness.

This paper introduced the key elements of BPILO, including:

- Parallel representation of business process models as process flow diagrams for interaction with users, and ontological models for computational reasoning
- Swarming unification of ontological process models with anytime, anywhere user interaction
- Comparison of processes using unified models, yielding visualizations of process similarities and differences, and metrics for process alignment.

Experimental results on model unification show the system working well to maximize the similarity of matched concepts and the resulting extracting of common definitional structure. Comparisons of the results of automatic unification to our manually unified model are promising but less convincing, apparently because of substantial holes in the representation of knowledge about the original models. These results lead us to near-term priorities for the research that include expanding the input data to include models of data and processes at multiple levels of detail, and full implementation of model unification including granularity adjustments that aggregate and sometimes split process concepts.

The key question for BPILO is whether model unification can be developed to a point where return on investment makes it clearly desirable for client organizations to participate in cross-organizational efforts to develop unified models. We know that manual unification of process models is possible, but expensive. We have shown that fully automatic unification yields results with some degree of face validity but also a lot of noise. We have an initial demonstration that suggests the potential for productive and pleasant anytime, anywhere user interaction with model unification. At this time we believe it is safe to conclude, as usual, that the BPILO research is promising and should be continued.

8 References

[BMMP 2005] BMMP Home Page (2005). DoD Business Transformation. http://www.defenselink.mil/bmmp/index.html.

[Daconta et al. 2003] Daconta, M. C., Obrst, L. J., Smth, K. T. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Wiley Publishing, Indianapolis, IN, USA.

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

[FEA 2005] Federal Enterprise Architecture (2005). http://www.whitehouse.gov/omb/egov/a-1-fea.html.

[IBM 2005] IBM Alphaworks (2005). Ontology-Based Web Services for Business Integration. http://www.alphaworks.ibm.com/tech/owsbi.

[K-CAP 2005] Integrating Ontologies Workshop (2005), Banff, Alberta. http://km.aifb.uni-karlsruhe.de/ws/intont2005/intontproceedings.pdf.

[OAEI 2005] Ontology Alignment Evaluation Initiative (2005). http://oaei.inrialpes.fr/2005/.

Knublauch, H., R. W. Fergerson, et al. (2004). The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. Third International Semantic Web Conference - ISWC 2004, Hiroshima, Japan (2004).

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

W3C Semantic Web (2001). http://www.w3.org/2001/sw/.

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