A Goal-based Model of Coordination in Interoperating Workflows

Vijay K. Vaishnavi*
GEORGIA STATE UNIVERSITY
Bill Kuechler
UNIVERSITY OF TEXAS AT SAN ANTONIO

January 4, 1999
GSU CIS Working Paper
CIS-99-4
A Goal-based Model of Coordination in Interoperating Workflows

Bill Kuechler
Accounting and Information Systems
University of Texas at San Antonio
E-mail: bkuechler@utsa.edu

Vijay Vaishnavi
Computer Information Systems
Georgia State University
E-mail: vvaishna@gsu.edu

Abstract

Automated workflow management systems (WFMS) between cooperating but autonomous groups can fail because autonomous workgroups are free to change the activities that constitute their task, disrupting coordination between systems. This paper presents a model of trigger-based coordination that is more robust than current WFMS implementations under conditions of unilateral process change.

1. Introduction

Most existing WFMS are based on ‘manufacturing models’ of work where precise specification of activities and their execution times are (presumed) critical. Yet many work processes, especially knowledge/office work, necessarily incorporate slack time to handle indeterminacy and frequently specify desired results intentionally, that is the process goal(s) are significant, but the details of execution are not [3]. The term intention has the same meaning in this paper as in intentional (transformational) programming [1]; that is, a high level specification of a desired system state that implicitly includes many (possibly an infinite number) of procedural paths to the goal state, termed extensions or instances.

The general problem of coordinating tasks across semantically heterogeneous environments constitutes a broad research area of current interest [4] [10]. Preserving coordination between environments that were initially semantically homogenous but that have undergone semantic drift [9] due to lack of communication and an autonomous relationship is an approachable subset of the general problem. We directly address some of

---

1 This work was partially supported by NSF Grants IIS-9810901 and IIS-9811248.
the semantic issues required to correct for semantic drift that have been identified by multiple authors [4] [10] as not adequately addressed by prior WFMS and process research. Though limited in scope, the repetitive production workflows addressed by this research constitute the majority of work performed at many organizations [8].

An earlier pilot study in this area [7] addresses the issues at a conceptual level. The research reported in this paper builds on the pilot study to develop a goal-based coordination model that has been implemented in a working prototype. The prototype computes semantic similarity between process definitions and uses this information to restore coordination by determining appropriate inter-process communication placement in redefined processes.

![Figure 1. Model of work definition](image)

2. Goal Based Coordination Model Framework

The cornerstone of the model is a work definition (WD) that specifies a work process as a hierarchy of goals for the activities of a work process (Figure 1). The set of goals at any level constitutes an intentional specification of all lower levels. Goals describe the what of a process, functions abstract the how, and activities describe a specific instantiation of a function. The root goal is the overall purpose of the work, and the name by which the work is commonly known in the organization. The layer of subgoals is generated during process design or specification, and represents a process of stepwise decomposition of the root goal into its components [12]. All subgoals except those in the final layer of subgoals are themselves goals for a lower level of description. When the subgoals are well enough defined to be implemented, they are linked to the generalized functionality by which they will be enacted. Incorporating the concept of functions into the WD more closely models the
manner in which workers conceive their activities than use of goals alone [11]. The terminal (leaf) nodes of the WD are the actual work activities containing predecessor / successor information. A method for "bottom up" explication of the goal / function hierarchy for existing work processes is given in [6].

Figure 2 shows the work definition instantiated for a simple test case that models the manufacture of a woman's suit, consisting of a vest and a skirt. Production of the vest is decomposed into sewing the vests and storing the vests until final packaging. The function that implements the storage subgoal is load&shipViaTruck, which is concretely realized by the activity vestsToWH3. The interprocess trigger, a communication to a trading partner to order cloth for other manufacturing processes (not shown) is linked to the completion of vestsToWH3. The linkage of a coordinating communication to a specific predefined activity state is common in WFMS and renders systems brittle under change. Following work redefinition (K5a -> K5b in Figure 2), the concrete activity has been changed to VestsTo9thStWH, resulting in coordination disruption and the failure to order the cloth in a timely manner.

To restore coordination, the trigger must be placed at the most appropriate location in the altered work definition; i.e. the most similar point in the work definition to the original trigger placement point must be determined. To accomplish this, a set of similarity heuristics based on Barsalou's work on ad-hoc categories [2] and Schanck's work on the relation of scripts and goals to understanding [11] are used. The key concept is that when activities and their goals are structured in a conceptual hierarchy as in our WDs, each higher level node
serves as a cognitive organizing principle for the nodes beneath it; this enables intentional classification and recognition of lower level nodes. These concepts are operationalized by two measures of similarity: intentional context (IC) and functional context (FC) (see Figure 3).

The path from the root goal to any activity node is termed the intentional context (IC) of the node, since tracing the path yields the semantics of the node. When ICs of activity nodes in different WDs are compared beginning at the root, the lowest identical node of the

![Image: Figure 3: Similarity Inferences Based on the Work Definition]

ICs is called the lowest common point (LCP) of the ICs. For example, if nodes 1 through 4 of WD's A and B in Figure 3 are identical, but node 5 differs in A and B then node 4 is the LCP of the ICs. The depth of the LCP is a measure of IC similarity. Conceptually the LCP is identical to what [1] terms the lowest common abstraction in the path from the goal node.

Activity nodes also have a functional context (FC), which is loosely defined as the set of activities adjacent to a given activity node. Functional context similarity between activity nodes i and x in two WDs is measured by an index ranging from 0 (no similarity) to 1 (identical). The index is computed by adding a weight to the index for each activity node position relative to i and x for which the nodes are identically defined. Weights diminish geometrically with distance from i and x, as shown in Figure 3.
FC and IC provide quantitative similarity measures for the assumptions and process-similarity heuristics [2] [11] that underlie the model. These can be expressed in two Axioms and three Principles:

**Axiom 1**: The root goals (the highest level goals for a work process) are identical for commensurable processes.

That is, in this model, if two work definitions can be compared, their root goals, the intentions of the overall work process, must be identical.

**Axiom 2**: Similar goals produce similar results, at the level of abstraction of the goal.

Any subprocess takes place in the context of an overall process, and any redefined version of a subprocess must achieve an equivalent result. Failure to meet this requirement would mean the failure of the overall process. In the running example, vests could be sewn in-house or by a subcontractor, by hand or by machine, but in any event, vests must be sewn or the overall process would fail to produce an outfit.

**Principle 1 (IC Similarity)**: The similarity of activity nodes i and x in two WDs increases with the depth of the LCP of the intentional contexts of i and x.

For example, if the nodes 1 through 5 in WDs A and B (Figure 3) were identical (LCP = 5), then even if activities i and x are described in terms unknown to the model, a rational inference would be that the activities were very similar. An LCP of lesser depth would still allow inferences of similarity to be drawn, but with less confidence.

**Principle 2 (FC Similarity)**: Similarity of activities increases with similar functional contexts.

**Principle 3**: Intentional context similarity (Principle 1) and functional contextual similarity (Principle 2) are additive.

If an activity placement for a trigger can be found that has high measures of both types of similarity, then the confidence in placement is increased.

Prior to discussion of the operation of these principles within a conceptual model of coordination, a brief discussion of the usage scenario envisioned for the model will provide a
being requested by the contractor, however the contractor requires notice of some intermediate point in the process (a trigger) to enable scheduling efficiencies. The coordination model attempts to determine the appropriate point in the altered definition to schedule the trigger, eliminating the user communication and reprogramming that would otherwise be required.

3. Coordination Model and Implementation

The conceptual model of coordination is shown in Figure 5. Work definitions (K5a, K5b) are input to a set of processes P1 through P3, which interpret them to produce the output (coordinated) WD (K5c). P4 provides high-level error handling. The initial comparison process (P1) analyzes the two work definitions and passes difference data to the recognition/interpretation process (P2). Using the similarity heuristics from the Section 2, P2 determines that point in the subcontractor definition that is semantically most similar to the original trigger placement point in the contractor definition. The scheduling process (P3) computes actual attachment nodes and performs the insertion of the trigger node in the
subcontractor WD. The lexicon (K6) is a simple dictionary for determining whether WD nodes are known to the subcontracting site.

3.2 Implementation. The prototype is an object-oriented system implemented in KAPPA, a PC version of KEE (Knowledge Engineering Environment). Its operation is presented next at an architectural level. The architecture of the prototype closely follows the conceptual model. The primary data entities of the conceptual model, work definitions (see Figure 6), difference data, and the lexicon are reified as architectural entities in the use case map of Figure 7 which traces flow through the prototype. Use case maps function similarly to object interaction diagrams but suppress inter-object message detail and so are more compact.

In the prototype the WDs represented in Figure 7 as single architectural entities are implemented as object structures linked by parent attributes in the class instances (nodes) (Figure 6).

The following discussion traces the activity of the prototype as it responds to the case illustrated in Figures 2 and 8. The solid directed line flowing through the entities represents control flow. Numbered text boxes describe prototype activity at each point. The prototype is quite general; preliminary processing that is required for all cases (not described here) detects
and completely handles situations for which no trigger rescheduling is required. This is the case when portions of process that do not communicate to external entities are modified.

When processing is required, it proceeds as shown in Figure 7. The Difference Data (DD) entity is the control nexus for this processing. After creating that entity (1), Work Request surrenders control to it. Using the principles discussed in Section 2, DD analyzes the contractor work definition to determine the intentional and functional context of the trigger (2). For the case in point, the intentional context is shown as the shaded path through the WD in process K5a of Figure 8. The functional context consists of those activities immediately surrounding vestsToWH3. Next, DD determines the most similar functional and intentional context in process K5c to that computed for K5a (3). For the case in point, the

Figure 6: A WD Viewed in the Prototype User Interface

Figure 7. Architectural trace of Trigger Reconciliation Processing
intentional and functional contexts for \textit{vestsToWH3} and \textit{vestsTo9ThStWH} are identical. The prototype interprets the situation as an instance of simple activity substitution (4), schedules the trigger to follow \textit{vestsTo9ThStWH} (5), and writes an explanation of its activities to an operator log (6) before returning to the host (7). The resultant WD is K5c in Figure 8.

4. Evaluation: Results and Conclusions

Logical analysis of the conceptual model developed in this paper (the basis for prototype implementation) shows it to handle a broad set of generic cases that address different kinds of coordination disruption as well as a human operating with equivalent information. The generic cases tested include substitution of a functionally equivalent activity (the case considered in this paper), change of function to accomplish a subgoal, and subgoal change accompanied by function and activity changes [5]. The analysis is confirmed by the exercise of the prototype with test cases. The model can be "fooled" by domain or site specific exceptions to the similarity heuristics, however the rule based design of the prototype enables the straightforward addition of high salience rules to accommodate those exceptions.

Future work is focused on (1) formalizing the conceptual model, and (2) expanding the scope of the model in a series of iterative refinement cycles to include multiple triggers and goal interactions and conflicts.
References


