Workflow Analysis Using Attributed Metagraphs

Amit Basu and Robert W. Blanning
Cox School of Business
Southern Methodist University
Dallas, TX 75275-0333
abasu@mail.cox.smu.edu

Robert W. Blanning
Owen Graduate School of Management
Vanderbilt University
Nashville, TN 37203
bob.blanning@owen.vanderbilt.edu

Abstract

Graph-theoretic methods for analyzing workflows and processes can provide insight into important problems in process design, and can help in the design of effective workflows. In particular, representation of workflows as metagraphs has been shown to provide a useful basis for formal analysis of processes and workflows. In this paper, we show how attributed metagraphs can be used to analyze workflows that have tasks with specified temporal constraints. In particular, we show how we can identify time-critical tasks and critical paths through workflows, which generalize traditional network scheduling (PERT/CPM) methods used in project management.

1. Introduction

Increasing competitive pressures have forced companies to review their operations in terms of business processes, which directly deliver value to the customer, rather than traditional functional departments such as engineering, manufacturing, and marketing [Davenport, 1993; Thompson, 1995]. Examples of such processes are order fulfillment, new product introduction, and loan processing. Since most business processes contain a large number of tasks, information elements, and resources, there is a need for formal methods for documenting and analyzing these processes, not only at the aggregate level, but also in terms of the individual workflows that instantiate the processes under specific conditions [Curtis, Kellner, and Over, 1992; Kwan and Balasubramanian, 1997; Tan and Harker, 1999].

Unfortunately, this is easier said than done. Most significant business processes in real companies involve large numbers of information elements (e.g., documents), tasks, and resources, both manual and automated. Furthermore, these components are scattered across multiple functional units with varied information management mechanisms, and even across multiple enterprises that constitute a virtual organization [Fleisch and Osterle, 2000]. Finally, each process includes several, and in some cases, many separate workflows (where a workflow is a particular instance of a process), depending upon conditions, assumptions and contexts [Hardwick and Bolton, 1997; Marshak, 1995]. The sheer scale and complexity of business process modeling and analysis motivates the need for formal methods of analysis.

In response to this need, there has been a growing interest in the areas of workflow management and process management, in both academia and industry. For instance, a number of computer-based packages for documenting and tracking workflows have been developed [Khoshafian and Buckiewicz, 1995]. There have also been efforts to standardize workflow technology, largely spearheaded by the Workflow Management Coalition [Swenson, 1995]. On the research front, there has been substantial progress in some areas of workflow analysis [Mackenzie, 2000], primarily on the transaction management aspects of workflows, where transaction models from database management systems have been extended to deal with processes that may be partially automated, and involve diverse resources.

In addition, there is a sizable research literature on workflow management, particularly on the transaction management aspects of workflow systems [Malone, Crowston, Lee, Pentland, Dellarocas, Wyner, Quimby, Osborn, and Bernstein, 1999]. The need for both graphical visualization tools for workflow monitoring and analytical tools for workflow analysis motivates the use of graph theoretic constructs for developing workflow management systems. While most existing tools and approaches use directed graphs for this purpose, the tasks in many processes involve set-to-set mappings for which directed graphs are unsuitable.

In recent years, a number of new graph theoretic constructs that are geared towards set-to-set mappings have been proposed. These include directed hypergraphs, higraphs, metagraphs and Petri nets. In particular, metagraphs [Basu and Blanning, 1994 a, b; 1997; 1998; 2000] have been shown to be effective for representing all the major components of business processes and their workflows (information elements/documents, tasks and resources) and for analyzing workflows using algebraic operations on several orthogonal views of workflows (document-centric, task-centric and resource-centric). However, the existing approach has focused on structural aspects of workflows – that is, the connectivity relationships among components of workflow systems. The goal of this paper is to extend the approach to include process attributes such as time.

In order to do this we must extend the existing literature on metagraphs to include the attachment of numerical attributes, which in this case will be time estimates, to metagraph edges. These will be estimates of the durations of the tasks represented by the metagraph edges. Other types of attributed metagraphs have been described in the literature, but the attributes have been logical (i.e., true/false) and qualitative, rather than quantitative (i.e., numerical). We will extend this literature to include quantitative attributes, which can be used to perform numerical calculations. This will allow analysis of scheduling aspects of workflows that are important for both design and ongoing management of business processes. As we will see, these aspects are in general analogous to but in detail different from the more familiar PERT/CPM approach to project management [Moder, Phillips, and Davis, 1983].

We begin in Section 2 by presenting basic definitions and describing basic properties of metagraphs, including the state of the art in attributed metagraphs. Then in Section 3 we will introduce the concept of numerically-attributed metagraphs and apply it to the scheduling of workflow systems.

2. Metagraphs

A metagraph is a graphical structure that represents directed relationships between sets of elements. The theory of metagraphs is available in [Basu and Blanning, 1994 a, b; 1997; 1998; 2000]. The remainder of this section is an overview of the basic features of metagraphs and also the state of the art in attributed metagraphs.

2.1 Basic Definitions

Given a finite generating set $X = \{x_i, =1...I\}$, a metagraph is an ordered pair $S = \langle X, E\rangle$ in which $E = \{e_k, k = 1...K\}$ is a set of edges. Each $x_i \in X$ is called an element. Each edge is an ordered pair $e_k = \langle V_k, W_k\rangle$ in which $V_k \subseteq X$ is the invertex of the edge e_k and $W_k \subseteq X$ is the outvertex of e_k. The coinput of any $x \in V_k$ is $V_k \setminus \{x\}$ and the cooutput of any $x \in W_k$ is $W_k \setminus \{x\}$.

An example of a metagraph is shown in Figure 1. This metagraph describes the activities and information elements in a loan processing business process. It contains four edges (labeled e_1,..., e_4) that relate various information elements of concern in terms of specific activities. The edge e_1 represents a credit check based upon the applicant's credit record (*Cust*) and the requested loan amount (*LAmt*). It results in a credit rating (*CR*) for the application. The edge e_2 represents a property appraisal activity, in which an appraised market value for the property under consideration (*AVal*) is computed, based upon the property record (*Prop*) and the requested loan amount. The edge e_3 represents an assessment of the bank's risk exposure (*RExp*) based upon the applicant's credit rating and the requested loan amount. The edge e_4 represents an approval procedure for the loan (*Appr*) based on the requested amount and the credit rating. Note that while some of these activities may be automated, others may be totally or partially manual.

Connectivity in metagraphs can be defined in terms of two types of paths, simple paths and metapaths. A simple path $h(x, y)$ is a directed sequence of connected edges linking two elements x and y. A cycle is a simple path $h(x, x)$ from any element x to itself. A metagraph containing at least one cycle is said to be cyclic, and a metagraph containing no cycles is said to be acyclic. On the other hand, a metapath $M(B, C)$ is a set of edges representing a metagraph whose pure inputs (i.e., the invertex elements that are not also outvertex elements) are in B and whose pure outputs (i.e., the outvertex elements that are not also invertex elements) contain C.

In general, given any source set B and target set C, there could be several metapaths from B to C. For example, in the metagraph in Figure 1, if $B = \{Cust, LAmt, Prop\}$ and $C = \{Appr\}$, then $\{e_1, e_4\}$ is a valid metapath. The possibility of multiple metapaths motivates two useful concepts, dominant metapaths and bridges. A metapath $M(B, C)$ is said to be dominant if there is no metapath to C from any proper subset of B and there is no proper subset $M' \subset M$ which is also a metapath from B to C. A bridge is an edge that occurs in every metapath between a given source and target. Thus, the edges e_1 and e_4 are bridges in the above example.

Metagraphs defined on the same generating sets can be combined, and a multiplication operator has been defined in terms of the adjacency matrix A of a metagraph, which can also be used to calculate the powers of an adjacency matrix. Each member of A^n, a^n_{ij}, is a set of zero or more triples, one for each simple path of length n connecting x_i to x_j. The first component of the triple is the coinput of x_i in the path, the second component is the cooutput of x_j, and the third component is the path.

Another useful construct is the closure A^* of the adjacency matrix, which is formed by adding successive powers of the matrix. Thus the closure is $A+A^2+A^3+A^4+...$ (at some point all successive A^n will have null elements and the summation can be terminated, since the multiplication operator does not allow more than one traversal of a cycle). Thus, the matrix A^* discloses all paths of any length connecting any two elements, and it can also be used to identify metapaths [Basu and Blanning, 1994(b)].

2.2 Attributed Metagraphs

As described thus far, metagraph edges are set-to-set mappings with no further information attached. However, it is possible to attach attributes to metagraph edges. We will describe three types of edge attributes. The first two, assumptions and resources, have been developed previously and have been reported in the literature. These are qualitative attributes, and they are described briefly in Section 2.2.1 below. The third type, numerical (or quantitative) attributes, is summarized in Section 2.2.2 and will be developed in more detail in this paper.

2.2.1 Qualitative Attributes

Two types of qualitative attributes in metagraphs are assumptions and resources. An assumption is a proposition – that is, a statement that may be either true or false [Basu and Blanning, 1998]. An assumption appears in the invertex of an edge and must be true for the edge to be used in a metapath. Each edge may contain zero, one, or more assumptions, and each assumption may appear in one or more edges. Propositions appear in the generating set along with elements representing other types of variables.

A conditional metagraph is a metagraph of the form $S = \langle X_p \cup X_v, E\rangle$, in which X_p is a set of propositions and X_v is a set of variables (i.e., the remaining elements). Note that a metagraph as defined in Section 2.1 above is a specialization of a conditional metagraph in which $X_p = \varnothing$.

The values of different propositions can be used to specify alternative contexts for a conditional metagraph. Thus, if a set of propositions P is true, another set Q is false, and the remaining propositions (i.e., $X_p \setminus (P \cup Q)$) are undetermined, then this knowledge can be used to simplify a conditional metagraph, so that only those edges that are valid in that context are retained. Specifically, the context of a conditional metagraph S with respect to P and Q, denoted $K(P, Q, S)$, is also a conditional metagraph in which (1) any proposition in P is deleted and (2) any edge containing a proposition in Q is deleted. If as a result any edge now has a null invertex or a null outvertex, then that edge is deleted as well. The resulting context metagraph will be a conditional metagraph because the undetermined propositions will remain as propositions in the context. The context operation is a useful abstraction on metagraphs, since it avoids the need to consider edges that cannot be used under the stated (Q) conditions.

Another type of qualitative attribute is a resource [Basu and Blanning, 2000]. Resources may be people, roles (i.e., classes of people, such as programmers or statisticians), and physical resources (e.g., workstations in general or a specific workstation). The resources associated with an edge must be present for the edge to appear in a metapath. The relationships between resources can be modeled by resource interaction metagraph, or RIM, which is formed as follows. In a RIM, the elements are resources. Therefore, the invertices and outvertices of the edges in a RIM are sets of resources. Assume that an outvertex of one edge in the original metagraph has a non-null intersection with the invertex of another edge in the original metagraph and that each of these two edges have resources associated with them. Then there is an edge in the RIM in which the resources of the first edge in the original metagraph are in the invertex and the resources of the second edge in the original metagraph are in the outvertex.

2.2.2 Quantitative Attributes

It is also possible to attach quantitative (numerical) attributes to metagraph edges. The purpose of this would be to allow certain calculations to be performed. For example, if the attributes are the costs of the tasks represented by the edges, then these attributes can be used to determine the total cost of the tasks appearing in a workflow. If the attributes represent the durations of the tasks, then they can be used to calculate the duration of the workflow in a fashion similar to the PERT/CPM calculations used in project management. If the attributes represent measures of performance, such as degrees of reliability or probabilities of non-failure, then they can be used to determine the performance of the workflow.

These attributes might also be combined. For example, if certain numerical attributes represent both time (i.e., activity durations) and cost, then these attributes might be combined to perform time/cost tradeoffs. If they represent either cost or duration along with probability of non-failure, then they might be used to determine the probability distributions of workflow cost or duration, depending on what will be done if a taskrepresented by and edge should fail. In this paper we will focus on deterministic activity durations, and we will not consider time/cost tradeoffs.

3. Critical Path Analysis of Workflows

In this section, we discuss how metagraph-based analysis of workflows can help identify critical activities and time-critical information elements in each workflow. This knowledge can then be used to manage the resources allocated to each activity in the workflow, and ultimately, in the underlying business process.

Consider a workflow metagraph in which each edge is labeled with its expected duration. An example of such a metagraph is shown in Figure 2. This is a modification of the example used in [Basu, Blanning, and Shtub, 1997]. The generating set consists of ten elements, each describing a document used in a life cycle costing workflow for a vehicle. There are five edges, each representing a task in the workflow. The edge e_1 represents a task that converts design specifications (*DSpec*) and production specifications (*PSpec*) into manufacturing cost estimates (*CEst*) and an estimate of service life (*ServL*). The edge e_2 represents a task that converts the vehicle's mileage report (*MRep*) yet another estimate of service life and also an estimate of annual operating cost (*ACst*). The edge e_3 represents a task that uses the manufacturing cost estimate and the company's markup policy (*MUPol*) to produce a sales price (*SalP*), and the edge e_4 uses the estimates of service life and annual operating cost to calculate the life cycle operating cost (*LCOC*). Finally, e_5 uses the sale price and the life cycle operating cost to calculate the total life cycle cost (*LCC*).

Given the source and target specification for this workflow, it is possible that the actual time available for some of the tasks is greater than the expected time, while other tasks are critical, in the sense that they have to be completed within very tight time constraints. We can identify the tasks and information elements in each category, using analysis similar to PERT/CPM methods in project management [Moder, Phillips, and Davis, 1983]. However, since the A^* matrix is available for the metagraph representation, it can be exploited in the process.

We treat cases where the outvertices of two or more edges overlap in a manner consistent with traditional project management approaches such as PERT/CPM. That is, when a particular information element is computed by multiple tasks in a workflow, its value is determined only after all these activities have completed. For example, in Figure 2, the value of *ServL* can be used as an input to another activity (e.g., activity e_4), only after both activities e_1 and e_2 have been completed. Given any acyclic metapath from a source set B to a target set C critical elements and activities can be found as follows:

Procedure Critical-Path (*M*, *B*, *C*)

<u>Phase 1: Early times</u>

For each element x_i in B, assign the label $Q_i = 0$, and mark x_i as live; let $Q_i = 0$ for all other elements.

Let $E = M(B, C)$

While $E \neq \emptyset$, for each edge e_j in E such that all elements in the invertex of e_j are live, do

Let $T_v = \max_{x_i \in Ve_j}(Q_i)$

For each $x_k \in W_{e_j}$, set $Q_k = \max[Q_k, (T_v + d_j)]$, and mark it as live if it is not already so.

Set $E = E \setminus \{e_j\}$

Repeat

Let $T^e = T^l = \max_{x_i \in X} Q_i$

<u>Phase 2: Late times</u>

Assign $L_i = T^l$ for all elements x_i in C, and mark them as live; set $L_i = \infty$ for all other elements.

Let $E_0 = M(B, C)$

While $E_0 \neq \emptyset$, for each edge e_j in E_0 such that all elements in the outvertex of e_j are live, do

Let $T_w = \min_{x_i \in We_j}(L_i)$

For each $x_k \in V_{e_j}$, set $L_k = \min[L_k, (T_w - d_j)]$ and mark it as live if it is not already so.

Set $E_0 = E_0 \setminus \{e_j\}$

Repeat

<u>Phase 3: Critical Elements</u>

For all $x_i \in X$, if $Q_i = L_i$ then x_i is marked as critical, with completion time $T_i = Q_i = L_i$

END.

Definition: An invertex V is <u>critical</u> if $\max_{x_i \in V}(Q_i) = \min_{x_i \in V}(L_i)$. We note that we will always have $\max_{x_i \in V}(Q_i) \le \min_{x_i \in V}(L_i)$ as long as Procedure Critical-Path is used to label the metagraph.

Theorem 1: An invertex V is critical if it contains any critical elements.

Proof: Without loss of generality, assume that V contains two elements, a critical element a and a non-critical element b. Let Q_a, L_a, and Q_b, L_b be the early times and late times respectively of these elements.

Then

$$\max_{x_i \in V}(Q_i) = \max[Q_a, Q_b], \min_{x_i \in V}(L_i) = \min[L_a, L_b]$$

Since V is feasible, it follows that $\max[Q_a, Q_b] \le \min[L_a, L_b]$

However, since a is critical, $Q_a = L_a$, and thus, $\max[Q_a, Q_b] \le \min[Q_a, L_b]$

There are three possible cases:

Case 1: $Q_a \le Q_b \le L_b$. In this case, $\max[Q_i] = Q_b$, $\min[L_i] = Q_a$ which makes the vertex infeasible unless $Q_b = Q_a$, in which case the vertex is critical.

Case 2: $Q_b \le Q_a \le L_b$. In this case, $\max[Q_i] = Q_a$ and $\min[L_i] = Q_a$, and the result follows.

Case 3: $Q_b \le L_b \le Q_a$. In this case, $\max[Q_i] = Q_a$ and $\min[L_i] = L_b$, and as in Case 1, the only feasible situation is when the vertex is critical.

QED.

We note that these elements must be contained in a critical invertex, except for those in the final outvertex, in this case **LCC**.

Definition: The slack in an edge *e* is defined as

$$slack(e) = \min_{x_i \in W_e}(L_i) - \max_{x_j \in V_e}(Q_j)$$

Definition: An edge is defined as critical if it has no slack.

Theorem 2: If the invertex of an edge *e* contains a critical element *a* with completion time T_a and the outvertex of the edge has a critical element *b* with completion time T_b such that $T_b - T_a = d_e$ then *e* is critical.

Proof: Since *a* and *b* are critical, it follows that $\min_{x_i \in W_e}(L_i) \le T_b, \max_{x_j \in V_e}(Q_j) \ge T_a$.

Thus, $\min_{x_i \in W_e}(L_i) - \max_{x_j \in V_e}(Q_j) \le T_b - T_a = d_e$, so that $slack(e) \le d_e - d_e = 0$, which proves the result.

QED.

Theorem 3: Each critical edge lies on a critical simple path from some element in *B* to some element in *C*.

Proof: Since the edges under consideration are all part of a metapath from *B* to *C*, each critical edge is on some simple path from an element in *B* to another in *C*. Now consider all such simple paths passing through a given critical edge *e*. If none of these paths is critical, then there is non-zero slack on all of them, which in turn means that *e* must have non-zero slack, which contradicts the claim that *e* is critical. Thus, at least one of these paths must have zero slack, and thus is critical.

QED.

We can illustrate these concepts and how they can be used, using the example workflow metagraph in Figure 2. Applying Procedure Critical-Path to this metagraph, we get a labeling of the early and late times for each element as illustrated in Figure 3. From Theorem 1, we can then infer that the critical vertices are {*DSpec*, *PSpec*}, {*CEst*, *ServL*}, {*SalP*, *LCOC*} and {*LCC*}. The critical edges in the workflow are then e_1, e_4, and e_5 (by Theorem 2) and thus the critical path through the workflow is $< e_1, e_4, e_5 >$ (by Theorem 3).

Once the critical path is known, scheduling and resource allocation of tasks can be done more effectively. For example, although task e_1 is critical, not all of its outputs are critical – that is, although the service life estimate (*ServL*) must be ready at the earliest possible time (i.e., time = 5), the manufacturing cost estimates (*CEst*) can be generated anytime between times 5 and 7. Thus, our analysis helps workflow managers not only with inter-task scheduling but also with intra-task scheduling.

These results can also be useful in resource allocation. The metagraph analysis enables managers to determine that any resources for tasks e_2 and e_3 can be reallocated as long as they are not critical – that is, as long as the reduced resources do not extend the task durations more than the existing slacks in the edges. Furthermore, even within critical tasks such as e_1 resources should be allocated to service life estimation with higher priority than to manufacturing cost estimation, unless other factors (e.g., quality or performance) become significant.

4. Conclusion

In this paper we have examined the problem of workflow scheduling. Specifically, we have studied the case in which the durations of the tasks in a workflow are known and we wish to calculate the duration of the entire workflow, along with any slack times for the tasks.

This is reminiscent of the problem of scheduling project networks using PERT/CPM. But there is an important difference. In project scheduling each activity has a single start event and a single end event. Thus, a digraph-based model is appropriate. However, in workflow systems tasks may have multiple inputs (typically hard-copy or electronic documents) and multiple outputs (also documents). One can "patch up" a digraph-based model with dummy events and tasks, but this would result in a clumsy model. Thus, a more sophisticated model is needed.

The graphical structure we applied to this problem is metagraphs, in which the edges are not ordered pairs of elements, as in digraphs, but ordered sets of elements. In other words, a metagraph is a collection of ordered set-to-set mappings, rather than a collection of ordered point-to-point mappings, as in a digraph. This allows us to model workflow tasks with multiple inputs and outputs.

Metagraphs, as studied thus far, are certainly adequate for modeling the qualitative aspects of workflows, such as determining whether certain source information is sufficient to produce certain target information. However, this view of metagraphs does not in itself provide a computational framework for workflow scheduling. In order to enhance metagraphs for this purpose, it is necessary to attach quantitative attributes (i.e., activity durations) to the edges representing the workflow tasks.

Attributed metagraphs have been studied in the past, but these have been qualitatively-attributed metagraphs. The qualitative attributes have been logical attributes (i.e., assumptions) and resources. Logical attributes appear in invertices and resources are attached directly to edges. The logical attributes must be true and the resources must be present for the edges to appear in a metapath. Quantitative, or numerical, attributes are different. These attributes are attached directly to edges, but they do not determine whether edges can appear in a metapath. Rather, they are used to perform calculations. The calculations considered in this paper are the time calculations needed for workflow scheduling.

There are two ways in which this work might be extended. The first is to modify the scheduling requirements appearing in the model. For example, we assumed that whenever an element appears in two or more outvertices it cannot be used in an invertex until all of the edges having these outvertices have been completed. In the example of Figures 2 and 3, e_4 cannot begin until both e_1 and e_2 have been completed, because ServL is in the outvertices of both e_1 and e_2 and is in the invertex of e_4. This is a conservative strategy. It is based on the notion that all methods of determining the value of an element should be completed so that the separate values can be compared or combined before proceeding.

Another possibility is to allow subsequent calculations to proceed as soon as any method for determining the values of the invertex elements has been completed. In other words, once a value of an element has been determined, it can be used in other tasks, even if there are still additional ways of determining the value and these have not yet been completed. In this case e_4 could begin after either e_1 or e_2 have been completed. Thus, metagraphs provide a foundation for workflow scheduling, and different scheduling requirements can be invoked as needed. We should point out that the theoretical results presented here (i.e., Theorems 1 – 3) may not be valid in some of these other cases.

We note that alternative possibilities such as these can be considered only if a metagraph is used to model the workflow. If a digraph of the PERT/CPM type is used, then only one possibility can be considered. In this case it is the conservative strategy, since all paths leading into a vertex must be completed before any activities leading out of the vertex can be executed. Thus, metagraphs provide an additional flexibility in modeling workflows.

The second extension would be to include other types of numerical attributes, such as cost, performance, and quality. Some of these might be straightforward. For example, the total cost of a set of tasks will be the sum of the costs of the individual tasks. The only graph-theoretic basis for this analysis is a hierarchical tree structure (sometimes called a Work Breakdown Structure) for classifying the costs. For example, assume that the workflow illustrated in Figures 2 and 3 takes place in an organization containing two departments: a Cost and Price Analysis Department containing the tasks e_1, e_2, and e_3 and a Life Cycle Estimating Department containing the tasks e_4 and e_5. Assume that the costs associated with these activities are $C(e_1) = 20$, $C(e_2) = 35$, $C(e_3) = 47$, $C(e_4) = 94$, and $C(e_5) = 83$. Then the cost for the Cost and Price Analysis Department would be 102, the cost for the Life Cycle Estimating Department would be 177, and the total workflow cost would be 279.

This type of thinking could be extended even further. For example, time/cost tradeoffs for the separate workflow tasks could be combined in a metagraph-based framework to yield time/cost tradeoffs for the entire workflow. Or a metagraph-based framework might be used to calculate probabilistic measures of time, cost, or performance (e.g., the probability of non-failure) for a workflow from individual probabilistic measures for the tasks in the workflow. Measures of performance or quality are quite another matter, especially since interactions among tasks may affect overall quality or performance in complicated ways. But it is possible that metagraph models of task interactions will be helpful in this more difficult area of workflow analysis.

In summary, metagraphs provide a powerful and flexible tool for scheduling workflows. The fundamental results in metagraph theory – such as the definition and construction of metapaths, the algebraic structure of metagraphs, and the construction of metagraph views – form a foundation for workflow analysis [Basu and Blanning, 2000]. By building on this foundation to include numerically-attributed edges, one can undertake scheduling and possibly other numerically-based efforts as well.

5. References

[1] Barua, Anitesh, Lee, C. H. Sophie, and Whinston, Andrew B., "The Calculus of Reengineering," *Information Systems Research*, Vol. 7, No. 4, December 1996, pp. 409-428.

[2] Basu, Amit and Blanning, Robert W., "Model Integration Using Metagraphs," *Information Systems Research*, Vol. 5, No. 3, September 1994(a), pp. 195-218.

[3] Basu, Amit and Blanning, Robert W., "Metagraphs: A Tool for Modeling Decision Support Systems," *Management Science*, Vol. 40, No. 12, December 1994(b), pp. 1579-1600.

[4] Basu, Amit and Blanning, Robert W., "Analysis of Assumptions in Model Bases Using Metagraphs," *Management Science*, Vol. 44, No. 7, July 1998, pp. 982-995.

[5] Basu, Amit and Blanning, Robert W., "A Formal Approach to Workflow Analysis," *Information Systems Research*, Vol. 11, No. 1, March 2000, pp.17-36.

[6] Basu, Amit, Blanning, Robert W., and Avraham Shtub, "Metagraphs in Hierarchical Modeling,"

Management Science, Vol. 43, No. 5, May 1997, pp. 623-639.

[7] Curtis, B., Kellner, M. I., and Over, J., "Process Modeling," *Communications of the ACM*, Vol. 35, No. 9, September 1992, pp. 75-90.

[8] Davenport, Thomas H., Process Innovation: Reengineering Work Through Information Technology, Harvard Business School Press, Boston, 1993.

[9] Davidow, William H. and Malone, Michael S., *The Virtual Corporation*, HarperCollins, New York, 1992.

[10] Fleisch, Elgar and Osterle, Hubert, "A Process-oriented Approach to Business Networking," E*lectronic Journal of Organizational Virtualness*, Vol. 2, No. 2, March 28, 2000, pp. 1-21.

[11] Hardwick, Martin and Bolton, Richard, "The Industrial Virtual Enterprise," *Communications of the ACM*, Vol. 40, No. 9, September 1997, pp. 59-60.

[12] Khoshafian, Setrag and Buckiewicz, Marek, *Introduction to Groupware, Workflow, and Workgroup Computing*, Wiley, New York, 1995.

[13] Kwan, M. M. and Balasubramanian, P. R., "Dynamic Workflow Management: A Framework for Modeling Workflows," *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, Vol. IV, January 1997, pp. 367-376.

[14] Mackenzie, Kenneth D., "Processes and Their Frameworks," *Management Science*, Vol. 46, No. 1, January 2000, pp. 110-125.

[15] Malone, Thomas W., Crowston, Kevin, Lee, Jintae, Pentland, Brian, Dellarocas, Chrysanthos, Wyner, George, Quimby, John, Osborn, Charles S., Bernstein, Abraham, Herman, George, Klein, Mark, and O'Donnell, Elissa, "Tools for Inventing Organizations: Toward a Handbook of Organizational Processes," *Management Science*, Vol. 45, No. 3, March 1999, pp. 425-443.

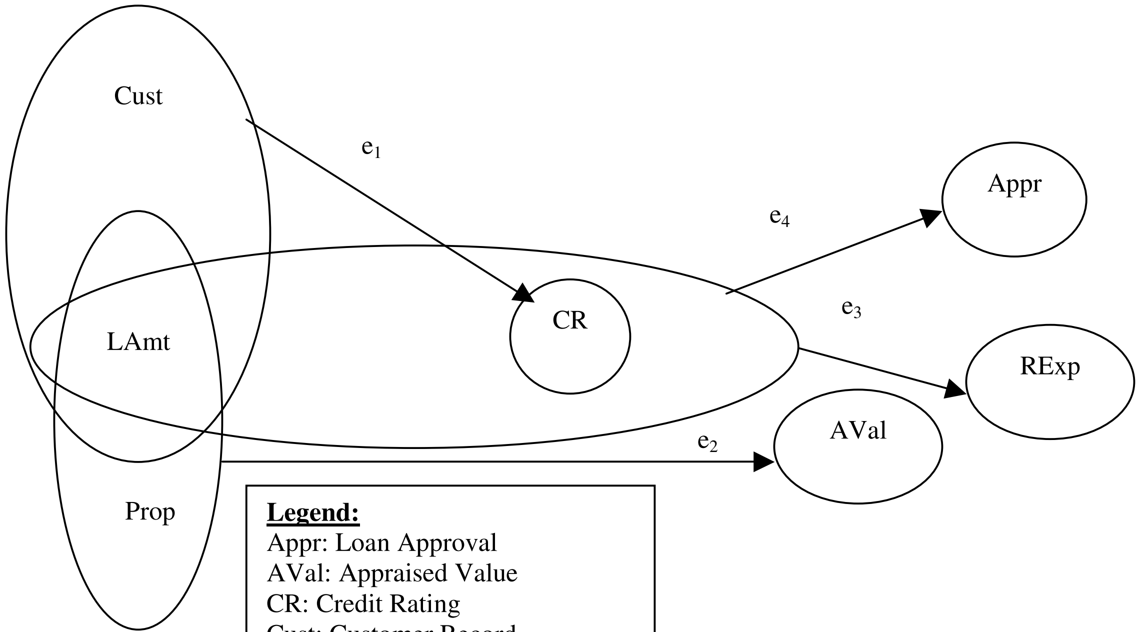
[16] Marshak, Ronni T., "Workflow: Applying Automation to Group Processes," Chapter 3 of *Groupware: Technologies and Applications*, ed. by David Coleman and Raman Khanna, Prentice Hall, Upper Saddle River, 1995, pp. 71-97.

[17] Moder, Joseph J., Phillips, Cecil R., and Davis, Edward W., *Project Management with CPM, PERT, and Precedence Diagramming*, 3rd ed., Van Nostrand Reinhold, New York, 1983.

[18] Swenson, Keith D., "Workflow Management Standards and Interoperability," in *New Tools for New Times: The Workflow Paradigm*, 2nd ed., ed. by Layna Fischer, Future Strategies, Inc., 1995, pp. 25-36.

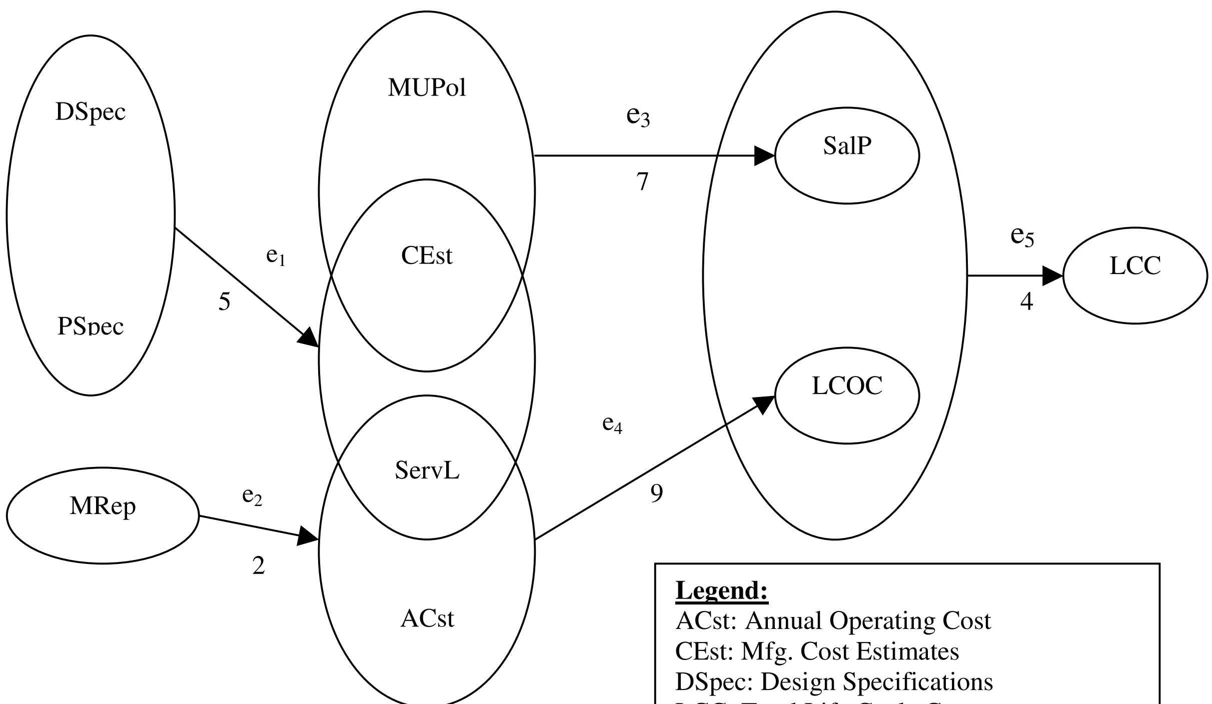
[19] Tan, Jui Chiew (J.C.) and Harker, Patrick T., "Designing Workflow Coordination: Centralized Versus Market-Based Mechanisms," *Information Systems Research*, Vol. 10, No. 4, December 1999, pp. 328-342.

[20] Thompson, V. J., "Process Monitoring for Continuous Process Improvement," in *Reengineering the Enterprise*, ed. by J. Browne and D. O'Sullivan, Chapman and Hall, London, 1995.



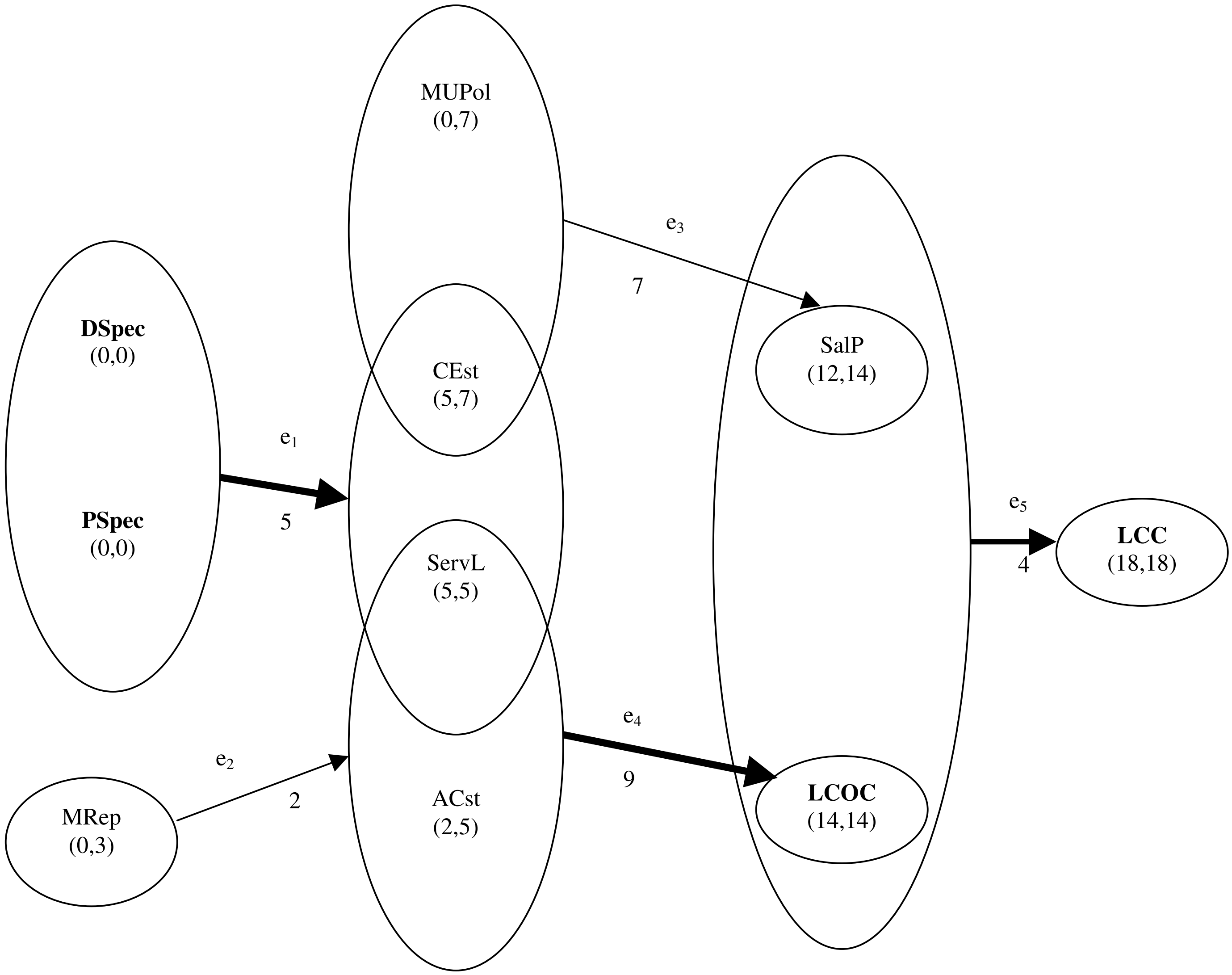
Legend:
Appr: Loan Approval
AVal: Appraised Value
CR: Credit Rating
Cust: Customer Record
LAmt: Loan Amount
Prop: Property Record
RExp: Risk Exposure

Figure 1: An Example Metagraph



Legend:
ACst: Annual Operating Cost
CEst: Mfg. Cost Estimates
DSpec: Design Specifications
LCC: Total Life Cycle Cost
LCOC: Life Cycle Operating Cost
MRep: Mileage Report
MuPol: Mark-up Policy
PSpec: Production Specifications
SalP: Sales Price
ServL: Service Life Estimate

Figure 2: Time-Constrained Workflow Metagraph



Note: The numbers in parentheses beneath each element x_i denote (Q_i, L_i) for that element, the thick arrows denote the critical edges, and boldface denotes critical elements.

Figure 3: The WF metagraph in Figure 2 with critical elements and critical path