

Tham, D., Fox, M.S., and Gruninger, M., (1994), "A Cost Ontology for Enterprise Modelling", Proceedings of the Third Workshop on Enabling Technologies - Infrastructures for Collaborative Enterprises, West Virginia University.

A COST ONTOLOGY FOR ENTERPRISE MODELLING

K. D.(Donald) Tham, Mark S. Fox, Michael Gruninger

Department of Industrial Engineering, University of Toronto
4 Taddle Creek Road, Toronto, Ontario, Canada, M5S 1A4

Abstract

There is an urgent need to formalize Activity-Based Costing (ABC) for purposes of implementation and usage in enterprises so that enterprises have access to the critical element of global success, viz., strategic management accounting. To make this possible, the authors present a core cost ontology and micro-theory of costing for enterprise modelling that spans the knowledge representation of activity, status of activity, time, causality, and resources. This ensures that ABC may be generically deployed in any enterprise to achieve effective activity-based cost management irrespective of the enterprise belonging to the manufacturing or service sectors.

Area: Cost Ontology, Activity-Based Costing (ABC), Strategic Cost Management, Enterprise Modelling, Business Process Reengineering, Enterprise Integration Framework.

1.0 Introduction

There is an urgent need to formalize Activity-Based Costing (ABC) for the purpose of incorporating costs into enterprise information systems. The September, 1993 issue of the FOCUS publication of the National Center for Manufacturing Sciences explicitly recognizes that ABC offers the management accounting strategy for American manufacturers wanting to achieve a strong global competitive advantage.

The ABC concept includes the assignment of cost to activities based on their use of resources, and the assignment of costs to "*cost objects*" based on their use of activities (CAM-I glossary, 1991). [Within the ABC literature, the term "*cost objects*" refers to the reasons for which activities are performed in enterprises]. *Since ABC assigns costs to activities based on their use of resources, the logical formulation of ABC must be premised on the existence of some given or identifiable basic or primitive resource costs, [later defined in Sec. 4.3 as Resource Cost Units], that must*

be associated with each resource that is required by an activity.

To ensure that ABC may be generically deployed in any enterprise (manufacturing or service industry), a core cost ontology and micro-theory of costing, that spans the representation of activity, status of activity, time, causality and resources, are presented for enterprise modelling. The deployment of ABC to improve business is commonly referred to as Activity-Based Management (ABM). ABC and ABM form the basis of strategic management accounting or activity-based cost management for global competitive advantage.

Establishing a core cost ontology guarantees the sharability and re-usability of appropriate terminology in communicating relevant cost data for ABC across all organizational departments. The activity centered costing micro-theory consists of axioms that use the cost ontology. These axioms have been developed in first order logic so as to provide the generic, sharable and re-usable mechanisms to compute and deduce costs when programmed in an AI language such as Prolog. Hence, our approach towards the formalization of ABC through the development of the cost ontology and micro-theory will ensure the minimum costs of software engineering in the computerization of activity-based cost management for the operational needs of the enterprise.

In the remainder of this paper, we describe the TOVE project, which provides the context in which our ontologies are being created. We then define TOVE's cost ontology followed by how it is to be used in cost management.

2.0 The TOVE Project and the Formalization of ABC

The TOVE Project, (TOronto Virtual Enterprise Project is a current ongoing project at the University of Toronto, Industrial Engineering Department), includes

two major undertakings: the development of an Enterprise Ontology, and a Testbed.

The TOVE Enterprise Ontology provides a generic, reusable ontology for modelling enterprises. An ontology is comprised of a reference data model composed of generic objects, attributes and relations, and formal definitions of terms and their constraints in First Order Logic. The TOVE ontology currently spans knowledge of activity, state, time, causality, resources, quality and cost (as developed for the formalization of ABC as per this paper). The ontology's data model is implemented on top of C++ using the Carnegie Group's ROCK (Representation of Corporate Knowledge) knowledge representation tool and the axioms are implemented in Quintus Prolog.

The TOVE Testbed provides an environment for analysing enterprise ontologies. The Testbed provides a model of an enterprise (a lamp manufacturing plant), and tools for browsing, visualization, simulation and deductive queries.

A micro-theory is a formalization of knowledge to perform a specific task. Our Activity-Based Costing micro-theory provides a logical formalization of the knowledge used to derive activity-based costs.

Firstly, the formalization of ABC through the development of *the cost ontology and costing micro-theory as developed in this paper is an extension and forms part of the TOVE Enterprise Ontology for enterprise modelling in general*. Secondly, some preliminary testing of the implementation of a limited cost ontology and micro-theory for ABC computations and deductability have been achieved through the usage of Quintus Prolog on the TOVE Testbed. By taking this approach towards the formalization of ABC, we hope to enhance the generic capabilities of enterprise modelling and enterprise engineering within the TOVE environment independent of the nature of the enterprise itself.

2.1 Time Representation in TOVE

Time is represented by points and periods (intervals) on a continuous time line. A time-point lies within an interval. A time-period is bounded by a start and end time-point. In TOVE, use of Allen's temporal relations [Allen 84] describe the relationships between time-points and/or time-periods.

FIGURE 1. Examples of Temporal Relations

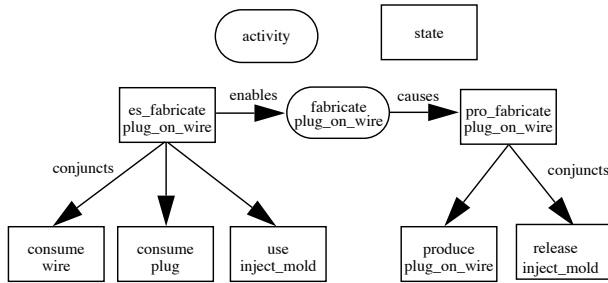
Relation	Symbol	Symbol for Inverse	Pictorial Example
X starts Y	s	si	
X before Y	<	>	
X during Y	d	di	
X ends Y	e	ei	

2.2 Activity/State Terminology and Semantics [Fox et al 93]

Enterprises are action oriented, and therefore, the ability to represent action lies at the heart of all enterprise models. The CIM-OSA model [ESPRIT 91] stratifies action from the lowest level of a *function*, to an *enterprise activity* and up to a *business process*; the Scheer representation [Scheer 89] defines function specific actions, and the PERA model [William 92] has a two level representation composed of a *task* at a lower level and a *function* at the upper level. In the CAM-I cost management system (CMS) model, a *function* is "a group of activities having a common objective within the business" [Berliner & Brimson 88]. In TOVE, a single entity called an activity spans all of the above. In this section, we briefly describe the terminology and semantics as per TOVE.

In TOVE, action is represented by the combination of an *activity* and its corresponding enabling and caused *states*. An activity is the basic transformational action primitive with which processes and operations can be represented. An enabling state defines what has to be true of the world in order for the activity to be performed. A caused state defines what will be true of the world once the activity has been completed. *An activity along with its enabling and caused states is called an activity cluster.*

FIGURE 2. Activity-State Cluster



An activity specifies a transformation on the world. Its status is reflected in an attribute called status. The domain of an activity's status is a set of linguistic constants:-

- **dormant** - the activity is idle and has never been executing before.
- **executing** - the activity is executing.
- **suspended** - the activity was executing and has been forced to an idle state.
- **reExecuting** - the activity is executing again.
- **completed** - the activity has finished.

2.3 Resource and State Terminology and Semantics [Fadel & Fox, 94]

“Being a resource” is not an innate property of an object, but is a property that is derived from the role an object plays with respect to an activity.

Hence, *the resource ontology includes the concepts of a resource being divisible, quantifiable, consumable, reusable, a component of, committed to, and having usage and consumption specifications.*

A *state* in TOVE represents what has to be true in the world in order for an activity to be performed, or what is true in the world after an activity is completed. States associate resources with activities through the *four types of states which reflect the four ways in which a resource is related to an activity - use, consume, release, produce.*

The status of a state, and any activity, is dependent on the status of the resources that the activity uses or consumes. All states are assigned a status with respect to a point in time. There are four different status predicates:-

- **committed** - a unit of the resource that the state *consumes or uses* has been reserved for consumption.
- **enabled** - a unit of the resource that the state *consumes or uses* is being consumed.
- **disenabled** - a unit of the resource that the state *consumes or uses* has become unavailable.

- **reenabled** - a unit of the resource that the state *consumes or uses* is re-available.
- **completed** - unit of the resource that the state *consumes or uses* has been consumed or used and is no longer needed.

3.0 Competency of the Cost Ontology

A problem in the development of ontologies is determining whether they are “correct”. A number of criteria have been proposed for evaluating ontologies, including: *generality, efficiency, perspicuity, transformability, extensibility, granularity, scalability* and *competence* [Fox & Tenenbaum 90] [Fox et. al. 93], and clarity, coherence, extensibility, minimal encoding and minimal and ontological commitment [Gruber 93].

The competence of a representation defines the types of tasks that the representation can be used in. The obvious way to demonstrate competence is to define a set of questions that can be answered by the ontology. If no inference capability is to be assumed, then question answering is strictly reducible to “looking up” an answer that is represented explicitly. In defining a shared representation, a key question then becomes: should we be restricted to just a terminology? Should the terminology assume an inheritance mechanism? Artificial Intelligence knowledge representations and object-oriented representations assume at least inheritance as a deduction mechanism. Or should we assume that some type of theorem proving capability is provided, say, in a logic programming language with axioms restricted to Horn clauses (i.e., Prolog)? What is the *deductive capability* that is to be assumed by an ontology? We propose that for each category of knowledge, a set of questions be defined that the ontology can answer. Given a representation and an accompanying theorem prover (i.e., Prolog), questions can be posed in the form of queries to be answered by the theorem prover. Given that a theorem prover is the deduction mechanism used to answer questions, the *efficiency* of a representation can be defined by the number of LIPS (Logical Inferences Per Second) required to answer a query.

Following are the questions we have identified as determining the competency of a cost ontology:-

1. What is the instantaneous and cumulative cost of a resource used in an activity a at time t?
2. What is the instantaneous and cumulative cost of an activity a at time t?
3. What is the instantaneous and cumulative cost of an order o at time t?
4. What is the cumulative cost of the class of activities a?

5. What is the cumulative cost of the class of orders o?

4.0 Cost Ontology for TOVE

We define *cost* as that entity which represents the temporal fiscal or monetary dimension, attribute, or characteristic of an enterprise activity, and may be referred to as *activity cost*.

In TOVE, costs will only change whenever the status of states, (hence resources), and activities change. Hence, we consider a cost event occurring when the status value of a state and activity change. The quantification of cost is treated constant with time until a status change occurs. In fact, we are applying situational calculus to model costs in a continuous dynamically changing world through the representation of discrete number of states of the world.

In order to reason about activities and their costs, we require a precise representation of:-

1. the generic classification or taxonomies of activity costs that intuitively or rationally present themselves in common sense enterprise modelling and our cost management perspective with TOVE;
2. the computational aspects in quantifying the types of activity costs.

Towards this end, the cost ontology for TOVE will serve as the data dictionary for the discourse domain of **the cost advisor agent - a software to communicate cost related data, to perform and to deduce cost computations that assist a human decision maker to accomplish ABC management at the generic level of enterprise modelling.** *The cost ontology uses the terminology and semantics of activity, state, resource and time that have been defined at a generic level for the TOVE environment [TOVE 93].*

4.1 Resource Cost Point of Activity, a, for Resource, r, at Time point, t: cpr(a,c,t,r)

The quantification of activity cost or the cost value of an activity associated with a required resource in TOVE is specified through the usage of the resource_cost_point predicate, cpr. The cost point predicate specifies the fiscal quantification of an activity's resource that requires a specified resource upto a certain instance of time. The resource_cost_point of activity for resource, r, at time point, t, in monetary units, c, is denoted by cpr(a, c, t, r,).

Definition: The resource_cost_point predicate, cpr, specifies the cost_value, c, (monetary units) of a

resource, r, required by an activity, a, upto a certain time point, t.

If a resource of the terminal use or consume states, s, for an activity, a, are enabled at time point, t, there must exist a cost_value, c, at time point, t, for the activity, a, that uses or consumes the resource, r. The time interval, $t_i = [t_s, t_e]$, during which a resource is used or consumed by an activity is specified in the use or consume specifications as use_spec(r, a, t_s, t_e, q) or consume_spec(r, a, t_s, t_e, q) where activity, a, uses or consumes quantity, q, of resource, r, during the time interval $[t_s, t_e]$. Hence,

Axiom 1: $\forall a, s, r, q, t_s, t_e, (use_spec(r, a, t_s, t_e, q) \wedge enabled(s, a, t)) \vee (consume_spec(r, a, t_s, t_e, q) \wedge enabled(s, a, t)) \equiv \exists c, cpr(a, c, t, r)$

[The computation for the resource cost point is discussed later in Axiom 30].

Semantics: the cost point predicate, cpr, is a ground term with four arities:

- **a**: activity identity or name of activity.
- **c**: cost_value of activity at the specified time point.
- **t**: a specified instance of time.
- **r**: an identified resource used or consumed by activity, a.

Example: cpr(assemble_clip_reading_lamp, 120, 75, nut4)

The above example indicates the resource_cost_point for the activity, clip_reading_lamp_assembly, is of cost_value 120 monetary units at time point 75, for the resource, nut4.

4.2 Cost point of Activity, a, at Time point, t: cpa(a,c,t)

Definition: The cost point of activity predicate, cpa, specifies the aggregate cost_value, c, of the activity, a, at time_point, t, given that the activity, a, uses and/or consumes one or more resources at the same point in time, t. In other words, this cost point predicate, cpa, is obtained by the summation of the cost value argument c_i for the resource_cost_point predicate, cpr, for all resources used and/or consumed by the activity, a, upto time point t.

This cost_point_for_activity definition returns the summation of all cost values c_i over all resources required by the activity at a specific time point as indicated in the axiom schema:-

Axiom 2: For each activity, a, and $\forall c_1, c_2, c_3, \dots, c_n, r_1, r_2, \dots, r_n, t, \exists c, cpa(a, c, t) \equiv cpr(a, c_1, t, r_1) \wedge cpr(a, c_2, t,$

$$r_2) \wedge cpr(a, c_3, t, r_3) \wedge \dots \wedge cpr(a, c_n, t, r_n) \wedge c = c_1 + c_2 + \dots + c_n$$

[The computation of the cost point for activity is later discussed in Axiom 31].

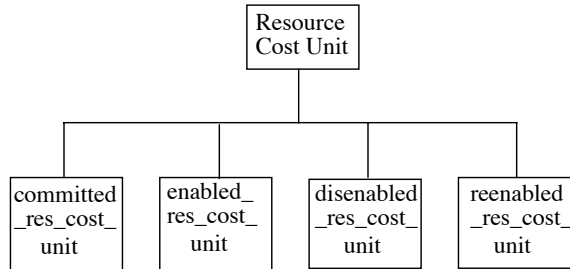
4.3 Taxonomy of Resource Cost Units

In TOVE, we define the basic or primitive cost_value of consuming or using 1 unit amount of a resource for 1 unit of time by an activity as the resource cost unit of the resource for the activity.

As per the status value of the use and/or consume terminal states of the enabling states of an activity, the activity status may be dormant, executing, suspended, reExecuting, or completed. Before being completed, an activity status may iterate through the various status values. The status of an activity depends on the status of the resources required by the activity.

In recognition of the above, our taxonomy of resource cost units consists of committed¹_res_cost_unit, enabled_res_cost_unit, disenabled_res_cost_unit and reenabled_res_cost_unit (see figure 3) where the nomenclature of the resource cost units are based on the state status values - committed, enabled, disenabled, reenabled - as was earlier defined in Sec.1.4 :-

FIGURE 3. Taxonomy of Resource Cost Units



Definitions pertaining to the resource cost unit classification or taxonomy are as follows:-

- dormant activity, a, requiring a resource, r, has a **committed_res_cost_unit**, (v_1), linking the resource and dormant activity through a cost metric, v_1 , expressed as \$/unit of resource/unit time;
- executing activity, a, requiring a resource, r, has an **enabled_res_cost_unit**, (v_2), linking the resource and executing activity through a cost metric, v_2 , expressed as \$/unit of resource/unit time;

1. Resources may be committed to a dormant activity.

- suspended activity, a, requiring a resource, r, has a **disenabled_res_cost_unit**, (v_3), linking the resource and suspended activity through a cost metric, v_3 , expressed as \$/unit of resource/unit time;
- reExecuting activity, a, requiring a resource, r, has a **reenabled_res_cost_unit**, (v_4), linking the resource and reExecuting activity through a cost metric, v_4 , expressed as \$/unit of resource/unit time.

We consider the above resource cost unit primitives v_1, v_2, v_3, v_4 as being the four cost attributes of the use and consume states associated with each resource specified for an activity. These resource cost units must be given or identifiable in the enterprise data model. [To ensure the completeness of the enterprise data model, refer to the closure axioms in the Appendix].

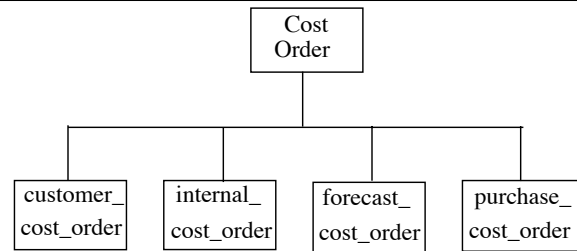
4.4 Taxonomy and Axioms for Cost Orders

Within the TOVE enterprise modelling paradigm, there are four generic and identifiable types of orders for which activities are performed at any given time point t. An activity is performed for a specific customer order, an internal order, a forecast order, or a purchase order. In other words, **an activity, a, at any time point, t, bears a relationship, viz., has_order, with the object, x, where x may be a customer order, an internal order, a forecast order, or a purchase order.** Hence, from a cost management standpoint, we accordingly classify these order types in the taxonomy of cost orders as shown in figure 4.

Expressing the definitions for the taxonomy of cost orders as axioms in first order logic would be as:-

Axiom 3: $\forall a, x, t, activity(a) \wedge has_order(a, x, t) \supset customer_cost_order(x, a, t) \equiv customer_order(x, t)$

FIGURE 4. Taxonomy of Cost Orders



Axiom 4: $\forall a, x, t, activity(a) \wedge has_order(a, x, t) \supset internal_cost_order(x, a, t) \equiv internal_order(x, t)$

Axiom 5: $\forall a, x, t, activity(a) \wedge has_order(a, x, t) \supset forecast_cost_order(x, a, t) \equiv forecast_order(x, t)$

Axiom 6: $\forall a, x, t, \text{activity}(a) \wedge \text{has_order}(a, x, t) \supset \text{purchase_cost_order}(x, a, t) \equiv \text{purchase_order}(x, t)$

4.5 Activity Cost Taxonomy and Axioms

From the preceding section, it is quite apparent that our cost ontology thus far has explicitly recognized the temporal behaviours of an activity and the cost of the activity that are closely associated with the status value of the enabling states and activity, together with the cost attributes of the use and consume terminal states in the enabling state tree for an activity.

Before an activity is completed, it is quite possible that the activity status value may have cycled through the dormant, executing, suspended and reExecuting status values. Hence, considering the taxonomy of resource cost units, and the temporal nature of activity status values, our activity cost profile must capture the cost of performing the activity dependent on its status value and resource cost unit allocation.

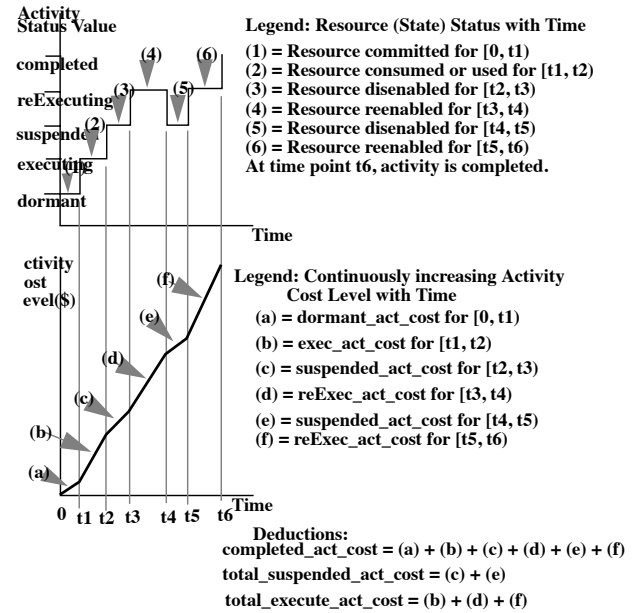
To be consistent and complete with the changing status of the activity, the temporal behaviour of costs may be captured through primitive activity cost terms identified as dormant_act_cost, exec_act_cost, suspended_act_cost, reExec_act_cost, and complete_act_cost.

Hence, in TOVE, we give explicit recognition to the temporal behaviour of activity cost by defining four primitive types of activity costs based on the status of the activity which depends on the status of the resources required by the activity, viz.,

1. dormant activity cost or *dormant_act_cost*,
2. executing activity cost or *execute_act_cost*,
3. suspended activity cost or *suspend_act_cost*, and
4. reExecuting activity cost or *reExec_act_cost*.

In general, from the above activity status and activity cost level profile (figure 5), an activity **holds** its status value for a time-period or time-interval with start point t and end point t' . During the interval (t, t') , the activity requires resource r . Hence, the cost allocation, c , for that activity during that interval is computed as the resource cost unit value, v , multiplied by the length of the time interval, $(t' - t)$, for which the activity holds a steady status value, multiplied by the quantity, q , of resource required by the activity during the time interval.

FIGURE 5. Resource (State) Status, Activity Status and Activity Cost Profile with Time



First we will define the intervals associated with the different values of status of a (resource) state:

Axiom 7: $\forall a, s, r, t, t', \text{committed_interval}(a, r, t, t') \equiv \forall t'', t \leq t'' < t' \supset \text{holds}(\text{committed}(s, a), t'') \wedge \neg \text{holds}(\text{committed}(s, a), t') \wedge \text{holds}(\text{committed}(s, a), t)$

Axiom 8: $\forall a, s, r, t, t', \text{enabled_interval}(a, r, t, t') \equiv \forall t'', t \leq t'' < t' \supset \text{holds}(\text{enabled}(s, a), t'') \wedge \neg \text{holds}(\text{enabled}(s, a), t') \wedge \text{holds}(\text{enabled}(s, a), t)$

Axiom 9: $\forall a, s, r, t, t', \text{disenabled_interval}(a, r, t, t') \equiv \forall t'', t \leq t'' < t' \supset \text{holds}(\text{disenabled}(s, a), t'') \wedge \neg \text{holds}(\text{disenabled}(s, a), t') \wedge \text{holds}(\text{disenabled}(s, a), t)$

Axiom 10: $\forall a, s, r, t, t', \text{reenabled_interval}(a, r, t, t') \equiv \forall t'', t \leq t'' < t' \supset \text{holds}(\text{reenabled}(s, a), t'') \wedge \neg \text{holds}(\text{reenabled}(s, a), t') \wedge \text{holds}(\text{reenabled}(s, a), t)$

Next, we compute the resource cost points for an activity at the endpoints of the status intervals.

Axiom 11: $\forall a, r, c, v, t, t', \text{committed_interval}(a, r, t, t') \supset [\text{committed_res_cost}(a, r, c, t') \equiv \text{committed_res_cost_unit}(a, r, q, v) \wedge c = v.(t' - t).q]$

Axiom 12: $\forall a, r, c, v, t, t', \text{enabled_interval}(a, r, t, t') \supset [\text{enabled_res_cost}(a, r, c, t') \equiv \text{enabled_res_cost_unit}(a, r, q, v) \wedge c = v.(t' - t).q]$

Axiom 13: $\forall a, r, c, v, t, t', \text{disenabled_interval}(a, r, t, t') \supset [\text{disenabled_res_cost}(a, r, c, t') \equiv \text{disenabled_res_cost_unit}(a, r, q, v) \wedge c = v.(t' - t).q]$

Axiom 14: $\forall a, r, c, t, t', \text{reenabled_interval}(a, t, t') \supset [\text{reenabled_res_cost}(a, r, c, t') \equiv \text{reenabled_res_cost_unit}(a, r, q, v) \wedge c = v.(t' - t).q]$

For time points that are during a status interval, the cost point of a resource at some time point t' has the same value as the cost point of the resource at the last time point t at which the status was changed. In order to express this, we need to say that the status was changed at time t and no action occurred between t and t' that could change the status.

Hence, if at time t the committed/enabled/disabled/reenabled resource cost point is of value c , and the status of the state for the activity is committed/enabled/disabled/reenabled at time t' , and the status of the state remains the same respectively between t and t' , then the resource cost point value, c' , of the activity at t' is the closest respective resource cost point value c of time t plus the cost of the resource used or consumed during the period $(t'-t)$. This is formalized in Axioms 15, 17, 19 and 21 respectively below. These axioms formalize the procedure of computing resource cost point values at t' as a cumulative cost upto t' and are increasing functions of the time period between t and t' .

On the other hand, for any time point t' beyond t , the committed/enabled/disabled/reenabled resource cost point value at t' is equivalent to the respective cost point value c at time t if the status of the state is non identical at t and t' and no event has occurred to change the status of the state at t . We therefore formulate the Axioms 16, 18, 20, and 22 respectively below for the resource cost points of each resource used or consumed by an activity.

Axioms 15 through 22 are formalized through the usage of the predicate `occursBet` (see **Appendix for Definition of occursBet**) as follows:

Axiom 15: $\forall a, r, c, c', t, t', \text{committed_res_cost}(a, r, c, t) \wedge \text{status}(s, a, \text{committed}, t') \wedge \neg \text{occursBet}(\text{commit}(s, a), t, t') \supset [\text{committed_res_cost}(a, r, c', t') \equiv \text{committed_res_cost_unit}(a, r, q, v) \wedge c' = c + v.(t' - t).q]$

Axiom 16: $\forall a, r, c, t, t', \text{committed_res_cost}(a, r, c, t) \wedge \neg \text{status}(s, a, \text{committed}, t') \wedge \neg \text{occursBet}(\text{commit}(s, a), t, t') \supset \text{committed_res_cost}(a, r, c, t')$

Axiom 17: $\forall a, r, c, c', t, t', \text{enabled_res_cost}(a, r, c, t) \wedge \text{status}(s, a, \text{enabled}, t') \wedge \neg \text{occursBet}(\text{enabled}(s, a), t, t') \supset [\text{enabled_res_cost}(a, r, c', t') \equiv \text{enabled_res_cost_unit}(a, r, q, v) \wedge c' = c + v.(t' - t).q]$

Axiom 18: $\forall a, r, c, t, t', \text{enabled_res_cost}(a, r, c, t) \wedge \neg \text{status}(s, a, \text{enabled}, t') \wedge \neg \text{occursBet}(\text{enabled}(s, a), t, t') \supset \text{enabled_res_cost}(a, r, c, t')$

Axiom 19: $\forall a, r, c, c', t, t', \text{disabled_res_cost}(a, r, c, t) \wedge \text{status}(s, a, \text{disabled}, t') \wedge \neg \text{occursBet}(\text{disabled}(s, a), t, t') \supset [\text{disabled_res_cost}(a, r, c', t') \equiv \text{disabled_res_cost_unit}(a, r, q, v) \wedge c' = c + v.(t' - t).q]$

Axiom 20: $\forall a, r, c, t, t', \text{disabled_res_cost}(a, r, c, t) \wedge \neg \text{status}(s, a, \text{disabled}, t') \wedge \neg \text{occursBet}(\text{disabled}(s, a), t, t') \supset \text{disabled_res_cost}(a, r, c, t')$

Axiom 21: $\forall a, r, c, c', t, t', \text{reenabled_res_cost}(a, r, c, t) \wedge \text{status}(s, a, \text{reenabled}, t') \wedge \neg \text{occursBet}(\text{reenabled}(s, a), t, t') \supset [\text{reenabled_res_cost}(a, r, c', t') \equiv \text{reenabled_res_cost_unit}(a, r, q, v) \wedge c' = c + v.(t' - t).q]$

Axiom 22: $\forall a, r, c, t, t', \text{reenabled_res_cost}(a, r, c, t) \wedge \neg \text{status}(s, a, \text{reenabled}, t') \wedge \neg \text{occursBet}(\text{reenabled}(s, a), t, t') \supset [\text{reenabled_res_cost}(a, r, c, t')]$

A resource may be disabled and reenabled several times before an activity is completed. Hence, there may be multiple intervals over which a state may be disabled and reenabled, and we must aggregate the costs for each of these intervals to compute the total cost for the disabled and reenabled status of the states. For this, we formulate the predicate `total_disabled_res_cost` and `total_reenabled_res_cost` to aggregate the costs upto time point t as follows:-

Axiom 23: $\forall r, a, t, c', c_1, c_2, \dots, c_k, t_1, t_2, \dots, t_n,$

$\text{total_disabled_res_cost}(a, r, c', t) \equiv [\text{disabled_interval}(a, r, t_1, t_2) \wedge \text{disabled_interval}(a, r, t_3, t_4) \wedge \dots \wedge \text{disabled_interval}(a, r, t_{n-1}, t_n)] \wedge [\text{disabled_res_cost}(a, r, c_1, t_2) \wedge \text{disabled_res_cost}(a, r, c_2, t_4) \wedge \dots \wedge \text{disabled_res_cost}(a, r, c_k, t_n)] \wedge [t_1 < t_2 < \dots < t_n \leq t] \wedge c' = c_1 + c_2 + \dots + c_n$

Axiom 24: $\forall r, a, t, c', c_1, c_2, \dots, c_k, t_1, t_2, \dots, t_n,$

$\text{total_reenabled_res_cost}(a, r, c', t) \equiv [\text{reenabled_interval}(a, r, t_1, t_2) \wedge \text{reenabled_interval}(a, r, t_3, t_4) \wedge \dots \wedge \text{reenabled_interval}(a, r, t_{n-1}, t_n)] \wedge [\text{reenabled_res_cost}(a, r, c_1, t_2) \wedge \text{reenabled_res_cost}(a, r, c_2, t_4) \wedge \dots \wedge \text{reenabled_res_cost}(a, r, c_k, t_n)] \wedge [t_1 < t_2 < \dots < t_n \leq t] \wedge c' = c_1 + c_2 + \dots + c_n$

However, a resource is committed and enabled only once to an activity before the activity is completed. Hence, the `total_committed_res_cost` and the `total_enabled_res_cost` for an activity at time point t' is equivalent to the `committed_res_cost` at t' and the `enabled_res_cost` at t' respectively.

Axiom 25: $\forall r, a, t, c, \text{total_committed_res_cost}(a, r, c, t) \equiv \text{committed_res_cost}(a, r, c, t)$

Axiom 26: $\forall r, a, t, c, total_enabled_res_cost(a, r, c, t) \equiv enabled_res_cost(a, r, c, t)$

An activity may use or consume n different resources. The cost of an activity being dormant, executing, suspended and reExecuting at time point t is computed by the aggregation of the total_committed_res_cost, the total_enabled_res_cost, the total_disenabled_res_cost, and the total_reenabled_res_cost at time point t respectively for each of the n resources that is used or consumed by the activity. We formalize these computations as per the following axioms:-

Axiom 27: For each activity, a, we have an axiom of the form:

$\forall c, c', r_1, r_2, \dots, r_n, t, dormant_act_cost(a, c', t) \equiv [total_committed_res_cost(a, r_1, c_1, t) \wedge total_committed_res_cost(a, r_2, c_2, t) \wedge total_committed_res_cost(a, r_3, c_3, t) \wedge \dots \wedge total_committed_res_cost(a, r_n, c_n, t)] \wedge c' = c_1 + c_2 + \dots + c_n$

Axiom 28: For each activity, a, we have an axiom of the form:

$\forall c, c', r_1, r_2, \dots, r_n, t, execute_act_cost(a, c', t) \equiv [total_enabled_res_cost(a, r_1, c_1, t) \wedge total_enabled_res_cost(a, r_2, c_2, t) \wedge total_enabled_res_cost(a, r_3, c_3, t) \wedge \dots \wedge total_enabled_res_cost(a, r_n, c_n, t)] \wedge c' = c_1 + c_2 + \dots + c_n$

Axiom 29: For each activity, a, we have an axiom of the form:

$\forall c, c', r_1, r_2, \dots, r_n, t, suspend_act_cost(a, c', t) \equiv [total_disenabled_res_cost(a, r_1, c_1, t) \wedge total_disenabled_res_cost(a, r_2, c_2, t) \wedge total_disenabled_res_cost(a, r_3, c_3, t) \wedge \dots \wedge total_disenabled_res_cost(a, r_n, c_n, t)] \wedge c' = c_1 + c_2 + \dots + c_n$

Axiom 30: For each activity, a, we have an axiom of the form:

$\forall c, c', r_1, r_2, \dots, r_n, t, reExecute_act_cost(a, c', t) \equiv [total_reenabled_res_cost(a, r_1, c_1, t) \wedge total_reenabled_res_cost(a, r_2, c_2, t) \wedge total_reenabled_res_cost(a, r_3, c_3, t) \wedge \dots \wedge total_reenabled_res_cost(a, r_n, c_n, t)] \wedge c' = c_1 + c_2 + \dots + c_n$

The cost point of an activity, cpa(a,c,t), at time t may be obtained as the sum of the dormant, execute, suspended and reExecute activity costs for the activity at the time t. Hence, the computation for the cost point value, c, for the activity, a, at time point, t, may be axiomitized as follows:-

Axiom 31: $\forall a, c_1, c_2, c_3, c_4, c', t,$

$cpa(a, c', t) \equiv total_dormant_act_cost(a, c_1, t) \wedge total_execute_act_cost(a, c_2, t) \wedge total_suspend_act_cost(a, c_3, t) \wedge total_reExec_act_cost(a, c_4, t) \wedge [c' = c_1 + c_2 + c_3 + c_4]$

The resource cost point of activity, a, for resource, r, at time point, t, cpr(a, c, t, r) is the sum of the total_committed_res_cost, total_enabled_res_cost, total_disenabled_res_cost and total_reenabled_res_cost at time point, t.

Axiom 32: $\forall a, c, c_1, c_2, c_3, c_4, t,$

$cpr(a, c, t, r) \equiv total_committed_res_cost(a, r, c_1, t) \wedge total_enabled_res_cost(a, r, c_2, t) \wedge total_disenabled_res_cost(a, r, c_3, t) \wedge total_reenabled_res_cost(a, r, c_4, t) \wedge [c' = c_1 + c_2 + c_3 + c_4]$

Further the cost point of an activity, a, at time point, t, may also be achieved as the aggregation of all resource cost points of all resources used or consumed by activity, a. Hence, as was stated in Axiom 2, we have

Axiom 2: For each activity, a, and $\forall c_1, c_2, c_3, \dots, c_n, r_1, r_2, \dots, r_n, t,$

$\exists c, cpa(a, c, t) \equiv cpr(a, c_1, t, r_1) \wedge cpr(a, c_2, t, r_2) \wedge cpr(a, c_3, t, r_3) \wedge \dots \wedge cpr(a, c_n, t, r_n) \wedge c = c_1 + c_2 + \dots + c_n$

[Note: Axiom 2 and Axiom 31 do give equivalent cost point value, c, for cpa(a,c,t)]

4.6 Activity Costs for Cost Orders

As stated in Sec. 4.4, we recognize four generic and identifiable types of cost orders for which activities are performed. These are the customer, internal, forecast and purchase cost orders. **It is reasonable to assign the costs of all activities to the cost order for which the activities were performed.**

Hence, we formulate the predicate, cpo, as the cost point of cost order, x, at time point, t, as the aggregate cost of all activities upto time point t for the cost order x.

Therefore, we axiomatize cpo(c', x, t) as the aggregation of cpa(a, c, t), the cost point of activities upto time point t as follows:-

Axiom 33: For each cost order, x, and $\forall c', a_1, a_2, \dots, a_n, c_1, c_2, \dots, c_n, t_1, t_2, \dots, t_n, t,$

$cpo(c', x, t) \equiv [has_cost_order(x, a_1, t_1) \wedge cpa(a_1, c_1, t_1)] \wedge [has_cost_order(x, a_2, t_2) \wedge cpa(a_2, c_2, t_2)] \wedge \dots \wedge [has_cost_order(x, a_n, t_n) \wedge cpa(a_n, c_n, t_n)] \wedge [t_1 < t_2 < \dots < t_n \leq t] \wedge [c = c_1 + c_2 + c_3 + \dots + c_n]$

5.0 Applying $cpr(a,c,t,r)$, $cpa(a,c,t)$ and $cpo(c,x,t)$ for Cost Management

The axioms developed thus far enable us to compute and deduce costs for an instance of an activity, a , and an instance of an order, x , or a specific order, x . Cost computations pertaining to an instance of an activity, a , and an instance of an order, x , have been achieved through our micro-theory (set of axioms) leading to the formulation of $cpa(a, c, t)$ and $cpo(c, x, t)$ respectively. Hence, thus far, we are able to provide answers to some of the following common sense queries:-

1. What is the instantaneous and cumulative cost of a resource used in an activity a at time t ?
2. What is the instantaneous and cumulative cost of an activity a at time t ?
3. What is the instantaneous and cumulative cost of an order o at time t ?

However, to enhance the application of our core cost ontology and micro-theory, we must extend our micro-theory of costs towards providing solutions to some of the further common sense queries put forth by enterprises wishing to achieve effective activity-based cost management:-

4. What is the cost of each subClass activity, a_i , when a_i has activity instances, a_{ij} 's?
5. What is the cost of Class_Activity, a'_{ix} , (eg. the class activity, Distribution) given that a'_{ix} (viz., Distribution) has subClass activities, a_i 's? (eg. Order-Picking, Palletizing, Material Handling, Shipping)? [Note: the subClass activity, Material Handling, may have activity instances like Hand Pallet Truck Handling, Fork-Lift Truck Handling, and ConveyORIZED Handling].
6. What is the cost of Cost_Order_Class, x_c , (eg. x_c may include all instances of orders that are fulfilled for the export sector or the electronic industry sector) given that x_c includes cost order instances, x_{ci} 's?

Computing and deducing costs for answers to queries 4, 5 and 6 involve the aggregation of costs at various activity levels (viz., activity instance, subClass activity, and Class_Activity) and order levels (viz., cost order instance, and Cost_Order_Class). Finding answers to such queries may be essential to achieve strategic cost management for multi-national, multi-subsidiary enterprises established or being established for the global market under trade alliances such as NAFTA and the European Union (EU). Hence, though queries 4, 5, and 6 are not meant to be totally exhaustive and mutually exclusive,

they serve as examples that strongly motivate the need to extend our cost ontology and micro-theory that involve the aggregation of costs through the various levels of activity and cost_order representations.

Figure 6 illustrates that activity classes, $a'_1, a'_2, a'_3, \dots, a'_n$ has_cost_order_class, x_c , upto time point, t . Activities $a_1, a_2, a_3, \dots, a_k$ are subClass activities of Class_Activity of a'_1 ; and $a_{i1}, a_{i2}, a_{i3}, \dots, a_{im}$ are instances_of a_i . Each instance, a_{ij} , of activity, a_i , uses/consumes resources $r_{i1}, r_{i2}, r_{i3}, \dots, r_{ip}$.

5.1 Computing cost point of subClass activity, a_i

From Axiom 32, for each resource, r_{ip} , required by a_{ij} , the resource cost point, cpr , is:-

$$\forall a, r, c', c_1, c_2, c_3, c_4, t,$$

$$cpr(a, c', t, r) \equiv total_committed_res_cost(a, r, c_1, t) \wedge total_enabled_res_cost(a, r, c_2, t) \wedge total_disabled_res_cost(a, r, c_3, t) \wedge total_reenabled_res_cost(a, r, c_4, t) \wedge [c' = c_1 + c_2 + c_3 + c_4]$$

From Axiom 2, the cost point of activity, a_{ij} , is the aggregation of cost point of resources, r_{ip} :-

$$For\ each\ activity\ instance,\ a_{ij},\ and\ \forall\ c, c_1, c_2, c_3, \dots, c_p, r_{i1}, r_{i2}, \dots, r_{ip}, t,$$

$$cpa(a_{ij}, c, t) \equiv cpr(a, c_1, t, r_{i1}) \wedge cpr(a, c_2, t, r_{i2}) \wedge cpr(a, c_3, t, r_{i3}) \wedge \dots \wedge cpr(a, c_p, t, r_{ip}) \wedge [c = c_1 + c_2 + \dots + c_p]$$

5.2 Computing cost point of subClass activity, a_i

The cost point of a subClass activity, a_i , is the aggregation of the cost point of each activity instance, a_{ij} . This computation may be axiomitized with the use of the distinguishing predicate, **cpa_subClass**, as follows:-

$$Axiom\ 34:\ For\ each\ subClass\ activity,\ a_i,\ and\ \forall\ c_i, t, a_{i1}, a_{i2}, \dots, a_{im}, c_{i1}, c_{i2}, \dots, c_{im},$$

$$instance_of(a_i, a_{i1}) \wedge instance_of(a_i, a_{i2}) \wedge instance_of(a_i, a_{i3}) \wedge \dots \wedge instance_of(a_i, a_{im}) \supset cpa_subClass(c_i, a_i, t) \equiv cpa(a_{i1}, c_{i1}, t) \wedge cpa(a_{i2}, c_{i2}, t) \wedge cpa(a_{i3}, c_{i3}, t) \wedge \dots \wedge cpa(a_{im}, c_{im}, t) \wedge [c_i = c_{i1} + c_{i2} + c_{i3} + \dots + c_{im}]$$

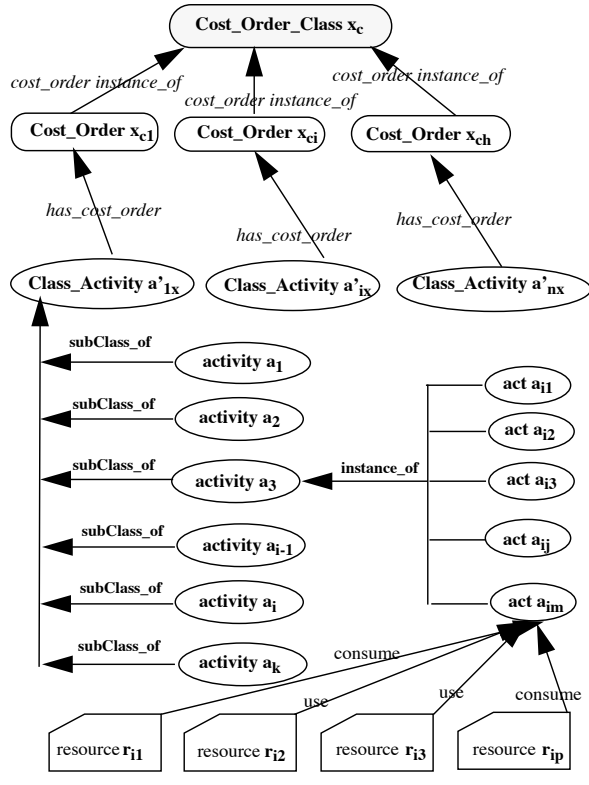
5.3 Computing cost point of Class Activity, a'_{ix} , required to satisfy cost order, x

The cost point of Class Activity, a'_{ix} , which is the i th class activity for cost order, x , is the aggregation of cost point of each subClass activity, a_i . This procedure is formalized as follows using the distinguishing predicate, **cpa_Class**, to indicate the cost point of an activity class:-

Axiom 35: For each Class activity, a'_{ix} , and $\forall c'_i, t, a_1, a_2, \dots, a_k, c_1, c_2, \dots, c_k$,

$$\text{subClass_of}(a'_{ix}, a_1) \wedge \text{subClass_of}(a'_{ix}, a_2) \wedge \dots \wedge \text{subClass_of}(a'_{ix}, a_k) \supset \mathbf{cpa_Class}(c'_1, a'_{ix}, t) \equiv \text{cpa}(a_1, c_1, t) \wedge \text{cpa}(a_2, c_2, t) \wedge \text{cpa}(a_3, c_3, t) \wedge \dots \wedge \text{cpa}(a_k, c_k, t) \wedge [c'_i = c_1 + c_2 + c_3 + \dots + c_k]$$

FIGURE 6. Relationships upto time point t amongst Cost Order Classes, Activity Classes, Activity Instances and Resources



5.4 Computing cost point of cost order, x

Given that the process plan for an order, x, has specified class activities, $a'_{1x}, a'_{2x}, \dots, a'_{nx}$, the cost point of x is the aggregation of the cost point of each class activity specified in the process plan of x. Thus, applying Axiom 33, we have the cost point of an order (or an instance of an order) as:-

For each cost order, x, and $\forall c'_x, a'_{1x}, a'_{2x}, \dots, a'_{nx}, c'_{1x}, c'_{2x}, \dots, c'_{nx}, t_1, t_2, \dots, t_n, t$,

$$\text{cpo}(c'_x, x, t) \equiv [\text{has_cost_order}(x, a'_{1x}, t_1) \wedge \text{cpa}(a'_{1x}, c'_{1x}, t_1)] \wedge [\text{has_cost_order}(x, a'_{2x}, t_2) \wedge \text{cpa}(a'_{2x}, c'_{2x}, t_2)] \wedge \dots \wedge [\text{has_cost_order}(x, a'_{nx}, t_n) \wedge \text{cpa}(a'_{nx}, c'_{nx}, t_n)] \wedge [t_1 < t_2 < \dots < t_n \leq t] \wedge [c'_x = c'_{1x} + c'_{2x} + c'_{3x} + \dots + c'_{nx}]$$

5.5 Computing cost point of cost order class, x_c

As illustrated in figure 6, cost order class, x_c , may be comprised of h number of cost order instances, $x_{c1}, x_{c2}, \dots, x_{ci}, \dots, x_{ch}$. Hence, each instance, x_{ci} , is a cost_order_instance_of cost order class, x_c . The costing of cost order class, x_c , is the aggregation of the cost order instances, x_{ci} 's. To formalize the aggregation of the cost order instances for the cost point of x_c , we use the distinguishing predicate, **cpo_class**, to indicate the cost point of cost order class, x_c as follows:-

Axiom 36:- For each Cost Order Class, x_c , and $\forall c_{xc}, t, x_{c1}, x_{c2}, \dots, x_{ch}, c_{x1}, c_{x2}, \dots, c_{xh}$,

$$\text{cost_order_instance_of}(x_c, x_{c1}) \wedge \text{cost_order_instance_of}(x_c, x_{c2}) \wedge \dots \wedge \text{cost_order_instance_of}(x_c, x_{ch}) \supset \mathbf{cpo_Class}(x_c, c_{xc}, t) \equiv \text{cpo}(c_{x1}, x_{c1}, t) \wedge \text{cpo}(c_{x2}, x_{c2}, t) \wedge \text{cpo}(c_{x3}, x_{c3}, t) \wedge \dots \wedge \text{cpo}(c_{xh}, x_{ch}, t) \wedge [c_{xc} = c_{x1} + c_{x2} + c_{x3} + \dots + c_{xh}]$$

6.0 Application of the TOVE Cost Ontology towards Activity Based Costing (ABC) System.

It is proposed that the usage and implementation of the foregoing core cost ontology in TOVE will enable companies to build an ABC system within the TOVE enterprise modelling paradigm, so as to provide enterprises the urgently needed "critical element to their global success - strategic management accounting" [FOCUS, *Strategic Management Accounting Offers American Manufacturers A Strong Global Competitive Advantage*, National Center for Manufacturing Sciences, Sept. 1993].

6.1 Conventional (Traditional) Cost Accounting Systems versus ABC Systems [Cooper 90]

Conventional cost accounting systems focus on units of particular products/services. Costs are allocated or "traced" to a product/service because *each unit of the product or service is assumed to consume resources*. Traditional allocation bases of resources to these units thus measure only attributes of a unit, eg. the number of direct labour hours, machine hours, or material costs consumed in making the product or providing the service. In accountancy terms, these allocation bases that measure characteristics of the product or service unit are called unit-level allocation bases.

However, in sharp contrast to the above, ABC systems focus on the activities performed to produce products or on the activities to provide a service. Costs are traced

from activities to products/services based on each product's or each service's consumption of the activities. Hence, *ABC acknowledges that products or services do not directly use up resources, but, instead, use up activities.* Consequently, the allocation bases, or "cost drivers", used in ABC are therefore measures of the activities performed.

6.2 Mapping the Conceptualization of ABC with the Cost Ontology [refer figure 7]

Resources are considered as the necessary requirements to accomplish or to perform an activity [Fadel & Fox 94]. In that sense, the property of the resource is dependent on the activity to be performed. Some examples of resources are machines, computers, materials, tools, humans, floor space, electricity, etc. However, *from a cost perspective, resources are the sources of cost and are viewed as economic elements directed to the performance of activities.*

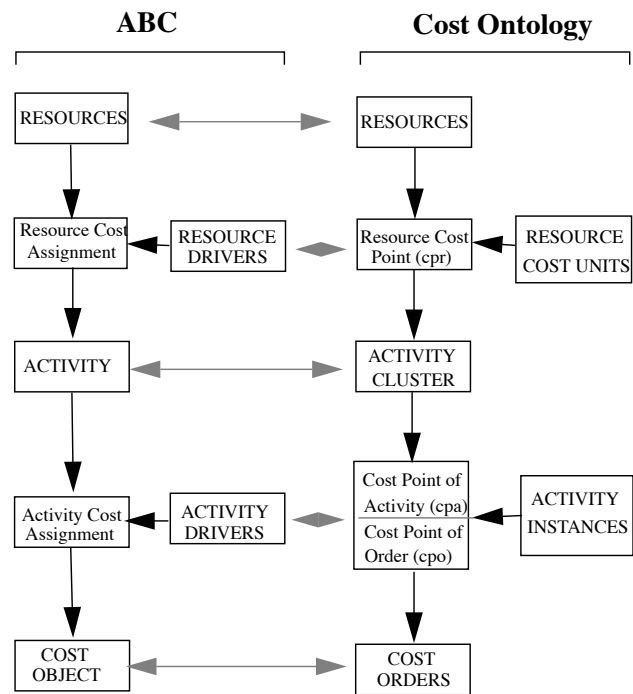
The *resource drivers* are "the links between the resources and activities. They take a cost from the general ledger and assign it to the activities"[Turney 92]. As many resources may be consumed or used by an activity, an activity may have several resource drivers. Looking for resource drivers, which are transaction-related "cost drivers", forms the first stage in cost management that helps management discover what contributes to costs [Stoffel 92]. *Our cost ontology enables the mapping of resource drivers to our resource cost units; whereas, the resource cost assignment of ABC is achieved through the cost micro-theory for the resource cost point (cpr) of an activity.*

In the ABC context, activity is considered "a combination of people, technology, raw materials, methods, and environment that produces a given product or service"[-Brimson 91]. The development of our cost ontology centers around the more precise and complete representation of the activity cluster.

In ABC, the reason for performing an activity is considered a cost object [Turney 92, ABC Glossary for CAM-I, Arlington, Texas]. A cost object is the reason why work is performed by an enterprise. Products and customers are reasons for performing activities. Cost objects include products, services, customers, projects and contracts. The cost object is the terminal point to which cost is traced. Consequently, the cost traced to each cost object will reflect the cost of the activities used by that cost object. *Our taxonomy of cost orders is a mapping of cost objects in ABC.*

In the ABC concept, each activity is traced to the cost object via an activity driver. *An activity driver is a measure of the consumption or usage of an activity by a cost object.* For example, the number of hours devoted by the design engineers to design a product may be considered as the activity driver for the engineering design activity.

FIGURE 7. Mapping the Conceptualization* of ABC (left side) with the Cost Ontology (right side)



*Based on [Turney 92] & [Turney 92]

ABC assigns the cost of activities to cost objects based on activity drivers that accurately measure consumption or usage of the activity. Cost objects are costed accurately when activity drivers measure the use of activities directly or correlate closely with their use. Hence, for the purpose of activity cost assignment to a cost object, the activity driver is used to assign resources from the activities to the cost objects. Identifying the most appropriate activity driver, which is also considered a transaction-related "cost driver", for an activity consumed by a cost object, forms the second stage of cost allocation in ABC that helps management discover what contributes to cost. Our representation of an activity cluster, together with the use or consume specifications of resources and the computations for the resource cost point of an activity, enable us to compute the cost point of an activity (cpa) and the cost point of a cost order (cpo) through the precise and complete repre-

sentation of an activity instance. Hence, *an activity instance in our cost ontology serves the purpose of assigning activity costs to an instance of a cost order just as an activity driver in ABC serves the purpose of activity cost assignment to a cost object.*

In summary, the conceptualization of ABC provides a framework to providing cost and operational information about the work carried on by the enterprise to be modelled. From the enterprise modelling perspective, it advantageously encompasses the following building blocks of the enterprise for their indicated purposes:-

- Resources for Resource Management,
- Resource Drivers for Cost Management and Business Process Re-design,
- Activities for Activity Management and Business Process Engineering,
- Activity Drivers for Cost Management and Business Process Re-design,
- Cost Objects for Strategic Management of the enterprise.

Therefore, the ABC framework of cost management for enterprise modelling points directly to profit opportunities by revealing the links from resource consumption to activities via resource drivers, and from activities to cost objects via activity drivers [Cooper & Kaplan 91].

The mapping of the ABC conceptualization to our developed cost ontology and micro-theory is evidenced by the following:-

1. Resource Drivers of ABC to our committed_res_cost_unit, suspend_res_cost_unit, execute_res_cost_unit, reExec_res_cost_unit;
2. Cost Objects of ABC to our Cost Orders in TOVE;
3. Activity Costs of ABC to our temporal and traceable dormant_act_cost, execute_act_cost, suspended_act_cost, reExec_act_cost;
4. The assignment of activity costs to instances of cost orders through the developed micro-theory involving resource cost point (cpr), cost point of activity (cpa) and cost point of order (cpo).
5. The aggregation of activity costs for a Class Activity and Cost Order Class through the extension of the cost ontology and micro-theory to include distinguishing predicates, cpa_subClass, cpa_Class, and cpo_Class, for the activity cost computations of cost point of subClass activity, cost point of Class Activity and cost point of Cost Order Class respectively.

7.0 Conclusions

This paper has described a core cost ontology and micro-theory of costing for enterprise modelling that spans the knowledge representation of activity, status of activity, time, causality, and resources. It has shown the mapping of the ABC concept to our cost ontology; and has put forth a cost micro-theory that makes it possible to reason, deduce and compute activity based costs for the operations of any enterprise. This would not only make possible the effective management of resources and activities towards an enterprise satisfying its clients, but would also provide an evaluation costing tool for business process design or re-design. Hence, our development should be considered a contribution towards fulfilling the urgent need of formalizing Activity-Based Costing (ABC) for purposes of implementation and usage in enterprises so that enterprises may attain global success through strategic management accounting.

The computations of activity based costs have been premised on the assumption that resource cost units are known or given for the enterprise modelled. The body of knowledge as to what contributes to the make up of resource cost units for an enterprise has not been definitively put forth. Hence, directions for future research should include the theory and body of knowledge to enable an enterprise to define and/or to deduce its resource cost units.

8.0 Appendix

8.1 Closure Axioms

Depending on the status value of the activity, each represented activity of a particular enterprise may require various resources. Hence, from a cost perspective, the following closure axioms expressed in first order logic are relevant to the particular enterprise being modelled so as to ensure that the resource cost units must be inputs as part of the data model that links each resource with each activity for the computations and deductions of activity costs towards the determination of resource cost points (cpr) of an activity, the cost point of an activity (cpa), and the cost point of an order (cpo).

Closure Axiom 1: $\forall a, r, q, v,$

$dorm_res_cost_unit(a,r,q,v) \equiv$

$(a = activity_1 \wedge r = resource_11 \wedge q = qty_11 \wedge v = value_11) \vee (a = activity_1 \wedge r = resource_12 \wedge q = qty_12 \wedge v = value_12) \vee (a = activity_1 \wedge r = resource_13 \wedge q = qty_13 \wedge v = value_13) \vee \dots \vee (a = activity_1 \wedge r = resource_1p \wedge q =$

$qty_1p \wedge v = value_1p) \vee \dots \vee (a = activity_n \wedge r = resource_n1 \wedge q = qty_n1 \wedge v = value_n1) \vee (a = activity_n \wedge r = resource_n2 \wedge q = qty_n2 \wedge v = value_n2) \vee (a = activity_n \wedge r = resource_n3 \wedge q = qty_n3 \wedge v = value_n3) \vee \dots \vee (a = activity_n \wedge r = resource_ny \wedge q = qty_ny \wedge v = value_ny),$

where n enterprise activities modelled, activity₁ requires p different resources and has committed resource cost unit values value₁₁, value₁₂, value₁₃,... value_{1p}, etc., respectively and activity_n requires y different resources and has y committed resource cost unit values value_{n1}, value_{n2}, value_{n3}, value_{ny}.

Closure Axiom 2: $\forall a, r, q, v,$

$exec_res_cost_unit(a,r,q,v) \equiv$

$(a = activity_1 \wedge r = resource_11 \wedge q = qty_11 \wedge v = value_11') \vee (a = activity_1 \wedge r = resource_12 \wedge q = qty_12 \wedge v = value_12') \vee (a = activity_1 \wedge r = resource_13 \wedge q = qty_13 \wedge v = value_13') \vee \dots \vee (a = activity_1 \wedge r = resource_1p \wedge q = qty_1p \wedge v = value_1p') \vee \dots \vee (a = activity_n \wedge r = resource_n1 \wedge q = qty_n1 \wedge v = value_n1') \vee (a = activity_n \wedge r = resource_n2 \wedge q = qty_n2 \wedge v = value_n2') \vee (a = activity_n \wedge r = resource_n3 \wedge q = qty_n3 \wedge v = value_n3') \vee \dots \vee (a = activity_n \wedge r = resource_ny \wedge q = qty_ny \wedge v = value_ny'),$

where n enterprise activities modelled, activity₁ requires p different resources and has p execute resource cost unit values value_{11'}, value_{12'}, value_{13'},... value_{1p'}, etc., respectively and activity_n requires y different resources and has y resource cost unit values value_{n1'}, value_{n2'}, value_{n3'}, value_{ny'}.

Similarly,

Closure Axiom 3: $\forall a, r, q, v,$

$suspend_res_cost_unit(a,r,q,v) \equiv$

$(a = activity_1 \wedge r = resource_11 \wedge q = qty_11 \wedge v = value_11'') \vee (a = activity_1 \wedge r = resource_12 \wedge q = qty_12 \wedge v = value_12'') \vee (a = activity_1 \wedge r = resource_13 \wedge q = qty_13 \wedge v = value_13'') \vee \dots \vee (a = activity_1 \wedge r = resource_1p \wedge q = qty_1p \wedge v = value_1p'') \vee \dots \vee (a = activity_n \wedge r = resource_n1 \wedge q = qty_n1 \wedge v = value_n1'') \vee (a = activity_n \wedge r = resource_n2 \wedge q = qty_n2 \wedge v = value_n2'') \vee (a = activity_n \wedge r = resource_n3 \wedge q = qty_n3 \wedge v = value_n3'') \vee \dots \vee (a = activity_n \wedge r = resource_ny \wedge q = qty_ny \wedge v = value_ny'');$

Closure Axiom 4: $\forall a, r, q, v,$

$reExec_res_cost_unit(a,r,q,v) \equiv$

$(a = activity_1 \wedge r = resource_11 \wedge q = qty_11 \wedge v = value_11''') \vee (a = activity_1 \wedge r = resource_12 \wedge q = qty_12 \wedge v = value_12''') \vee (a = activity_1 \wedge r = resource_13 \wedge q = qty_13 \wedge v = value_13''') \vee \dots \vee (a = activity_1 \wedge r = resource_1p \wedge q = qty_1p \wedge v = value_1p''') \vee \dots \vee (a = activity_n \wedge r = resource_n1 \wedge q = qty_n1 \wedge v = value_n1''') \vee (a = activity_n \wedge r = resource_n2 \wedge q = qty_n2 \wedge v = value_n2''') \vee (a = activity_n \wedge r = resource_n3 \wedge q = qty_n3 \wedge v = value_n3''') \vee \dots \vee (a = activity_n \wedge r = resource_ny \wedge q = qty_ny \wedge v = value_ny''')$

8.2 Definition of occursBet

The predicate occursBet is used to represent the fact that an action occurs between two time points t and t'.

Axioms 16, 17, 18, and 19 make use of the negation of the predicate occurBet to expressly state that, if no action occurs between t and t', and the resource cost point at t has value c, then the resource cost point at t' also has value c since no action has occurred between t and t'.

9.0 Acknowledgements

[This research is supported, in part, by The Natural Science and Engineering Research Council, Carnegie Group Inc., Digital Equipment Corp., Micro Electronics and Computer Research Corp., Quintus Corp., and Spar Aerospace.]

10.0 References

- [Allen 84] Allen, J.F. Towards a General Theory of Action and Time. *Artificial Intelligence*. 23(2):123-154, 1984.
- [Berliner & Brimson 88] Berliner, Callie & Brimson, James A., *Cost Management for Today's Advanced Manufacturing: the CAM-I conceptual design*, Free Press: New York, 1988.
- [Brimson 91] Brimson, James A., *Activity Accounting: an activity-based costing approach*, John Wiley & Sons, Inc., 1991.
- [Cooper 90] Cooper, Robin, ABC: A NEED, NOT AN OPTION, *Accountancy*, pp. 86-88, September, 1990.
- [Cooper & Kaplan 91] Cooper, Robin & Kaplan, Robert S., *Profit Priorities from Activity-Based Costing*,

- Harvard Business Review, pp.130-135, May-June, 1991.
6. [DoD 93] The Office of the Director of Defense Information (no author), The DoD Enterprise Model, Office of the Secretary of Defense, Washington D.C., USA, February 16, 1993.
 7. [ESPRIT 91] ESPRIT Consortium AMICE (no author), Open System Architecture for CIM, CIM-OSA AD 1.0, Architecture Description, (ESPRIT - Project 688, Project 2422, Project 5288), 2 Boulevard de la Woluwe Bte. 8, B-1150 Brussels, Belgium, 1991.
 8. [Fadel et al 93] Fadel, Fadi George, Fox, Mark S., Gruninger, M., A Resource Ontology for Enterprise Modelling, Submitted paper for the Knowledge Representation (KR) Conference, 1994.
 9. [Fadel & Fox 94] Fadel, Fadi George and Fox, Mark S., A Resource Ontology for Enterprise Modelling, Submitted paper for the Third Industrial Engineering Conference, 1994.
 10. [Fox & Tenenbaum 90] Fox, M.S., and Tenenbaum, J.M., Proceedings of the DARPA Knowledge Sharing Workshop, Santa Barbara CA, 1990.
 11. [Fox et al 93] Fox, Mark S., Chionglo, John F., Fadel, Fadi G., Towards Common Sense Modelling of an Enterprise, Proceedings of the Second Industrial Engineering Research Conference, 1993.
 12. [Hammer 90] Hammer, Michael, Reengineering Work: Don't Automate, Obliterate, Harvard Business Review, July-August, 1990.
 13. [Morrow & Hazell 92] Morrow, Michael & Hazell, Martin, Activity Mapping For Business Process Redesign, Management Accounting, pp. 36-38, February, 1992.
 14. [Motro 93] Motro, Amihai, Responding with Knowledge, International Journal of Expert Systems, Vol. 6, No.1, pp. 121-138, 1993.
 15. [Sathi 85] Sathi, A., Fox, M.S., and Greenberg, M., Representation of Activity Knowledge for Project Management, IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-7(5):531-552, September, 1985.
 16. [Scheer 89] Scheer, August-Wilhelm, Enterprise-Wide Data Modelling, Springer-Verlag, 1989.
 17. [Stoffel 92] Stoffel, Thomas J., Activity-based Costing: The Competitive Advantage for the 1990s, The Journal of Applied Manufacturing Systems, pp. 58-63, Winter 1992.
 18. [Tham 93] Tham, K. Donald, Cost Perspectives in Enterprise Modelling, A working paper for the Enterprise Engineering Laboratory under the directorship of Mark S. Fox, University of Toronto, August 31, 1993.
 19. [TOVE 92] Fox, Mark S., Chionglo, John F., Fadel, Fadi G., The Toronto Virtual Enterprise model, Enterprise Integration Laboratory, University of Toronto, 1993.
 20. [Turney 92] Turney, Peter B.B., Common Cents: The ABC Performance Breakthrough (How to succeed with activity-based costing), Portland, OR: Cost Technology, 1992.
 21. [Turney 92] Turney, Peter B.B., Activity-based Management, The Journal of Applied Manufacturing Systems, pp. 29-36, Winter 1992.
 22. [Williams 92] Williams, Theodore J., The Purdue Enterprise Reference Architecture, Purdue Laboratory for Applied Industrial Control, Purdue University, West Lafayette, Indiana 47907, USA, March 16, 1992.