Using Object Templates from the REA Accounting Model to Engineer Business Processes and Tasks.

Guido L. Geerts

Department of Accounting and MIS

University of Delaware, Newark, DE 19716-2715

geerts@aisvillage.com; 302/831-6413

William E. McCarthy

Department of Accounting and Information Systems

Michigan State University, East Lansing, MI 48824, USA

mccarth4@msu.edu; 517/432-2913

ABSTRACT: Conceptual models of enterprises can be used for both business process modeling and the actual design of computerized information systems.The Resource-Event-Agent (REA) model, a domain-specific framework for determining the information architecture of accounting and enterprise information systems, has primarily been used for design purposes.In this paper we explore the use of REA for business process modeling.At first, we discuss how REA primitives are used to describe the value-added transformation of resources throughout the enterprise. Next, we analyze some difficult specification problems that relate to business process modeling in general and to REA engineering solutions to these problems in particular.We conclude the paper with the specification of a three-layer architecture that summarizes the REA approach to business process modeling: (i) enterprise-level specification, (ii) REA-based process descriptions, and (iii) task-level or workflow specifications.The architecture shows that workflow descriptions can be related to the specification of the enterprise value chain through the medium of REA information structures.

KEY WORDS: Accounting information systems, business process modeling, enterprise value chain, REA model. workflow modeling.

1. INTRODUCTION

The discipline of enterprise information system engineering has dramatically changed in recent years.Two major innovations of significant import are briefly discussed below.

a. *Advances in information technology -- such as client-server architectures, object technology and a variety of Internet technologies such as XML -- are dramatically changing the way enterprise systems are designed, implemented, andoperated.*

The use of object-oriented technology has resulted in a decrease of the semantic gap between the analysis, design and implementation phases of information systems (Jacobson 1992, Walden and Nerson 1995). As a result, conceptually designed business models can be fully carried into implementation, and the links between different enterprises are much tighter. XML (extensible markup language) technologies promise to couple different enterprise systems together much more closely than they are historically accustomed to (Bradley 2000, Hoque 2000).

b. *Conceptual design methods have been assigned a dual role – the understanding of the business as well as the actual design of the enterprise information system* (Jacobson et al. 1995, Taylor 1995, Eriksson and Penker 2000).

Historically, conceptual models have been used primarily to capture information requirements and to translate these requirements smoothly into a variety of database and file-oriented platforms (such as network, hierarchical and relational systems).According to Batini et al. "*Conceptual design starts from the specification of requirements and results in the conceptual schema of a [particular] database."* (1992, p.6). More recently, Gale and Eldred (1996, p. 121) describe the new purpose of conceptual modeling as: "*to construct a model of any kind of domain using a set of theoretical modeling constructs that capture the way the world really is."*In a business environment, this world consists (among other things) of business processes and business rules.The practice of **B**usiness **P**rocess **R**eengineering or BPR (Hammer and Champy, 1993) requires accurate modeling and documentation techniques for business realities, and conceptual models turn out to be an excellent tool for these purposes.

These advances in technology and specification methods have also dramatically changed the applicability of a particular conceptual method -- the REA (**R**esource-**E**vent-**A**gent) model (McCarthy 1982, Geerts and McCarthy, 2000a).The REA model was introduced by McCarthy in 1982 as a domain-specific theory for the **design** of accounting information systems, and its use has gradually been expanded to the modeling of economic phenomena in general (Geerts and McCarthy, 1994; 2000a).A main characteristic of REA is that it heavily relies on first-order accounting principles. The use of REA for the actual design of information systems has been widely discussed in the literature (see for example: McCarthy (1982), Gal and McCarthy (1986), Hollander et al. (2000), Romney and Steinbart (2000), Hall (2001), and Geerts and McCarthy (2000a, 2000b, 2001)).Geerts and McCarthy (2000b) describe how knowledge-based technology enables transformation of REA from a mere design methodology into an **operational framework** for information systems.

In this paper, we focus on the concurrent use of REA for the design of an enterprise's conceptual information architecture and for business process modeling.We start with a brief discussion of Porter's enterprise value chain concept, and we relate that framework to business process engineering in general and to the REA model in particular.To illustrate the meaning of REA concepts, we next introduce a simple business example that is used throughout this paper: the **R**ent-**A**-**C**razy-**C**ar (RACC) Business. We illustrate an REA object template explosion for one of RACC's main business processes (maintenance), and we construct a simple REA-based value chain for RACC.The paper then analyzes some difficult specification problems that relate to

business process modeling in general and to REA engineering solutions to these problems in particular.With these specification problems discussed, the paper returns to its central BPM focus and shows how low-level, function-oriented information processing tasks can be related to high-level models of value-added behavior through the medium of REA information structures.We do this with examples drawn from the renting process of RACC. We end with some conclusions and further research directions.

2. ENTERPRISE VALUE CHAINS WITH RESOURCE-EVENT-AGENT COMPONENTS

Michael Porter's (1985) *Value Chain* concept "was developed as a systematic method for examining all the activities a firm provides and how they interact" (Callon, 1996, p. 47).As Callon (1996) notes, Porter's value chain idea in its original incarnation was used primarily as a source for analyzing competitive advantage.It did not include the notion of providing customer value as the ultimate objective, nor was it ever intended to be used as a framework for the information architecture of an enterprise.In actual practice however, Porter's product and service flows can be arranged as a series of input-output processes with resource flows between them (Geerts and McCarthy, 1994; 1999), which have the ultimate objective of customer value:

> A fundamental notion in value chain analysis is that a product gains value (and costs) as it passes through the vertical stream of production within the firm (design, production, marketing, delivery, service).When created value exceeds cost, a profit is generated.This notion of value creation is derived from the economics of demand.Products are viewed as bundles of attributes (Lancaster, 1975) which can be configured in multiple ways to appeal to segments of consumers having diverse demand functions (Hergert and Morris, 1989, p.183).

A value chain at its highest level of abstraction for a typical manufacturing company (Callon, 1996, p. 48) might include product and service flow through: (1) research and development, (2) engineering, (3) production and manufacturing, (4) marketing, (5) sales and distribution, and (6) service.Each step in this chain can be considered a *process* as Hammer and Champy (1993, p. 53) define it: "A collection of activities that takes one or more kinds of input and creates an output that is of value to the customer."

In an economic sense, a *process* is a production function wherein the entrepreneur or manager of an enterprise exchanges some input resource(s) for some output resource(s) of greater value to the customer.For example, in the third process shown in the paragraph above (production and manufacturing), the inputs would be labor, machines, and raw material which are converted into goods that have more value to the customers than the sum of the values of the inputs. In the fourth process (marketing), the inputs could be cash and manufactured goods which are then converted to advertised final goods (something of more value to the customer).

2.1. Decomposition of an enterprise value chain

The manufacturing company specified above would have six processes in its simplest and most abstract form.In a realistic information system engineering setting, most of these would need to be

decomposed further before actual process engineering would start.This is illustrated in the process hierarchy of figure 1 and explained below.

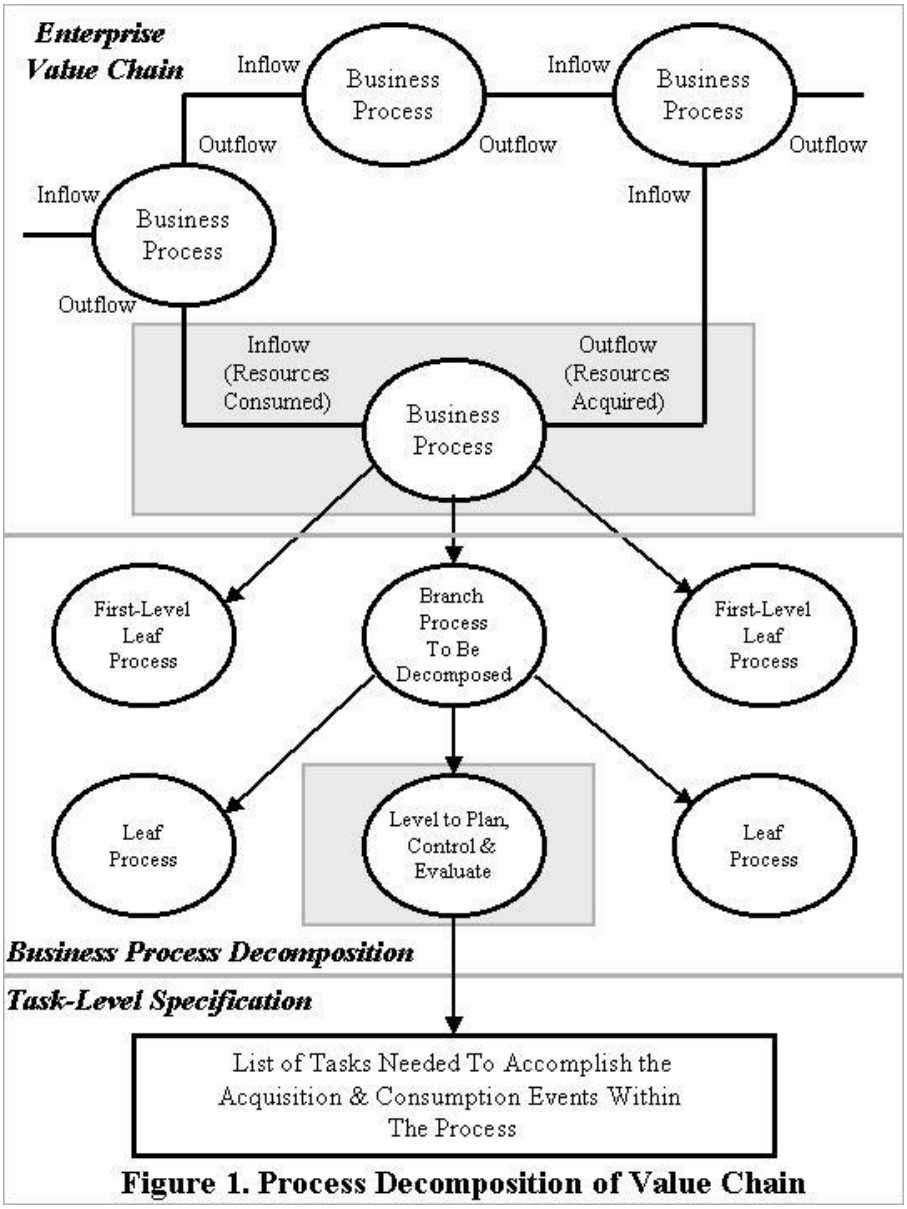

Figure 1. Process Decomposition of Value Chain

At the top of figure 1, an *enterprise value chain* is portrayed as a series of connected inflow-outflow processes. Each business process is adding value by converting resources into a more valuable (to customers) resource.The new resource is then used as input by another business process. The *business process decomposition* layer, in the middle of figure 1, illustrates further decomposition of one of the business processes in the enterprise value chain. The decomposition consists of two "leaf" nodes on either side of a middle process that is decomposed yet further.Let us suppose that the*production and manufacturing* process, which is part of the enterprise value chain, can be further decomposed into three sub-processes -- *set-up, assembly*, and *inspection* -- with the first and third of these being leaf processes (i.e., not needing further decomposition).Further, we may suppose that the sub-process *assembly* can be decomposed into

three more parts: *combining*, *welding*, and*painting*.This gives us an example for the three-level decomposition shown.

Each **process** in an enterprise value chain (at any level of abstraction or decomposition) has at least two composite **economic events**: a decrement event that consumes the inputted resource and an increment event that acquires the outputted resource.For example, a revenue process has a sale event (consumption of inventory input) and a cash receipt event (acquisition of cash output).As we shall see in the event template description below, the REA model provides a representation pattern for these events and for other objects within a process.At a later point in the paper, we will discuss situations where these increment/decrement events might be modeled at a conceptual level below the lowest process decomposition.

In any decomposition of enterprise business processes, a heuristic is needed for stopping the decomposition.The one we adopt here is adapted from Hollander et al. (2000) -- *conceptual modelers decompose from the enterprise-level value chain down until they reach the lowest level at which decision-makers need information: (1) to plan and design the operation and sequencing of activities of the future, and (2) to monitor the operation of activities of the present, and (3) to evaluate the operation ofactivities of the past.* Beyond this point, decision usefulness wanes and implementation technology details become overwhelming.In figure 1 we use this heuristic to stop further decomposition of the middle bottom process in layer 2. Any further specifications of a bottom level process are accomplished via task descriptions. Task-level or workflow specifications are portrayed as the bottom layer in figure 1, and specification of workflow occurrences is a point we return to later in the paper.

2.2. The REA object template

We defined a business process above as an entrepreneur exchange or transformation where input is changed into output with the express objective of providing customer value.Each process has at least one input consumption event and one output acquisition event. McCarthy's (1982) REA model is a semantic model for an economic exchange that pairs two mirror-image event object patterns, one of them a template for a give (a consumption event) and the other a take (an acquisition event).Geerts and McCarthy (2000a) further refine the semantic description of an economic exchange and we use their representation in the rest of this paper. An REA object template for a single process is illustrated in figure 2 and explained below.

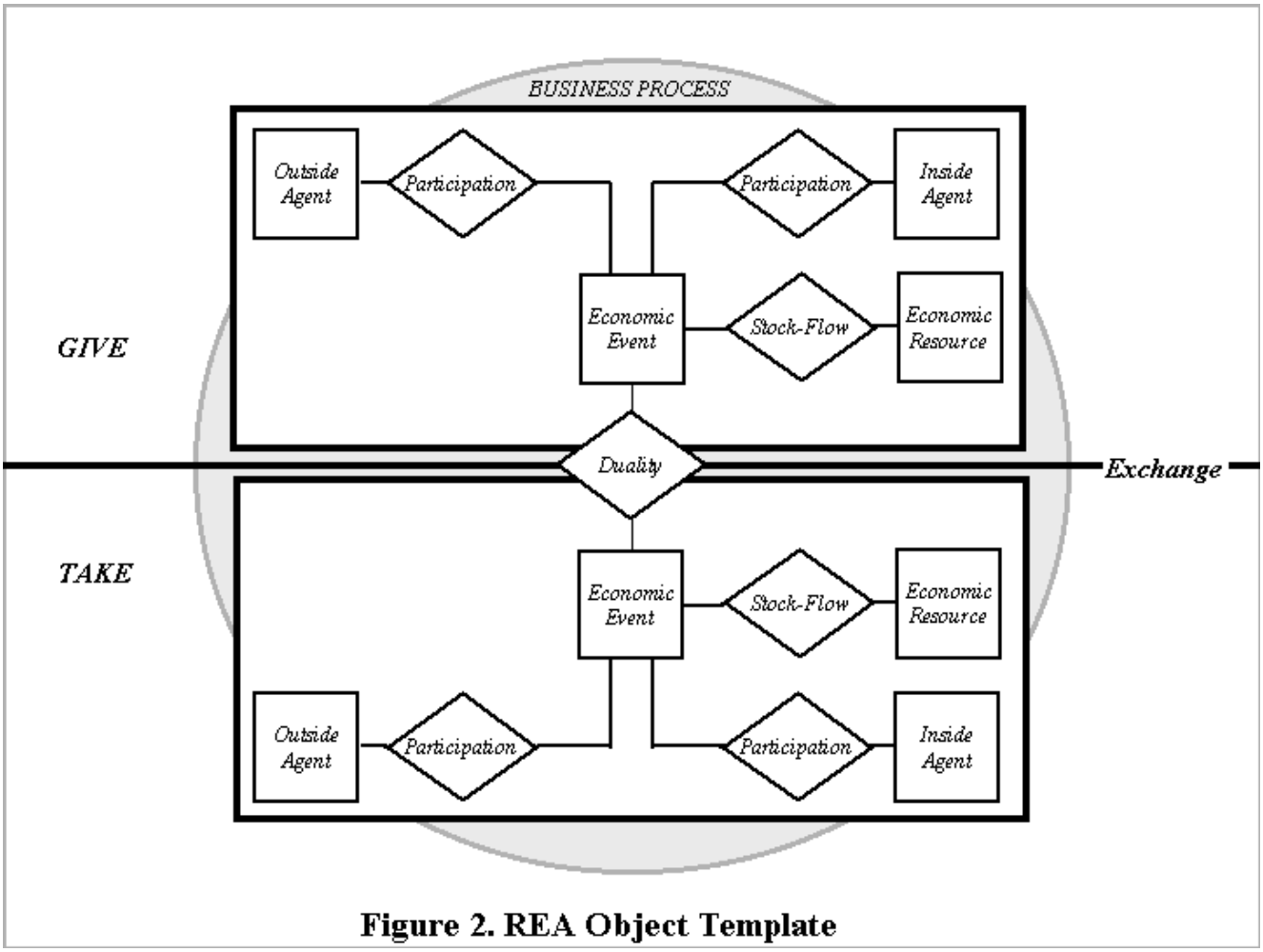

Figure 2. REA Object Template

Two of REA's primitives are *Economic Event* and *Economic Resource*.Economic Event is the central dynamic primitive of McCarthy's domain theory and is described by Yu (1976, p.256) as "*a class of phenomena which reflect changes in scarce means resulting from production, exchange, consumption, and distribution [processes]*."Economic Events are critical information elements of an enterprise information system that describe the inflow (consumption) and outflow (acquisition) of Economic Resources. Economic Resource is the central static primitive of the domain theory, and it describes the stock of resources kept at a certain point in time. Obviously, this stock is affected by Economic Events, and this relationship is reflected in the *Stock-Flow* primitive (figure 2).

Another REA primitive is the concept of *Exchange* or *Give-Take* relationship that is titled *Duality*in figure 2.Fisher (1906, p.149) describes an exchange (or transfer) as follows: "*transfers usually occur in pairs, and involve two objects transferred in opposite directions between two owners.This double transfer, we have called exchange."*

The remaining REA primitives are used to model the *Inside Agents* and *the Outside Agents*involved in the Economic Events.Inside Agents are usually employees accountable for the Economic Events, while Outside Agents are the external parties involved in the exchange. The

relationships connecting the Economic Event with the inside and outside Agents are named *Participation*relationships.[1]

If we combine the value chain model of figure 1 with the process object pattern of figure 2, we derive the extended REA model described by Geerts and McCarthy (1994; 1999).In actuality, this extended model has some other declarative and procedural primitives (such as the concept of resource-event-agent typification plus the concepts of commitments and claims), but a discussion of these is beyond the scope of this paper which focuses on process engineering.Interested readers may consult Geerts and McCarthy (2000a).

A full economic data model for an enterprise may be derived by exploding each of the leaf node processes of its decomposed value chain into its REA components.We will demonstrate this procedure in the context of a specific example in the next section of the paper.

2.3. The Rent-A-Crazy-Car (RACC) example

Figure 3 shows the result of applying the REA object template to one of the business processes (maintenance) of our business example: **R**ent-**A**-**C**razy-**C**ar (RACC). We will explain RACC's business first.

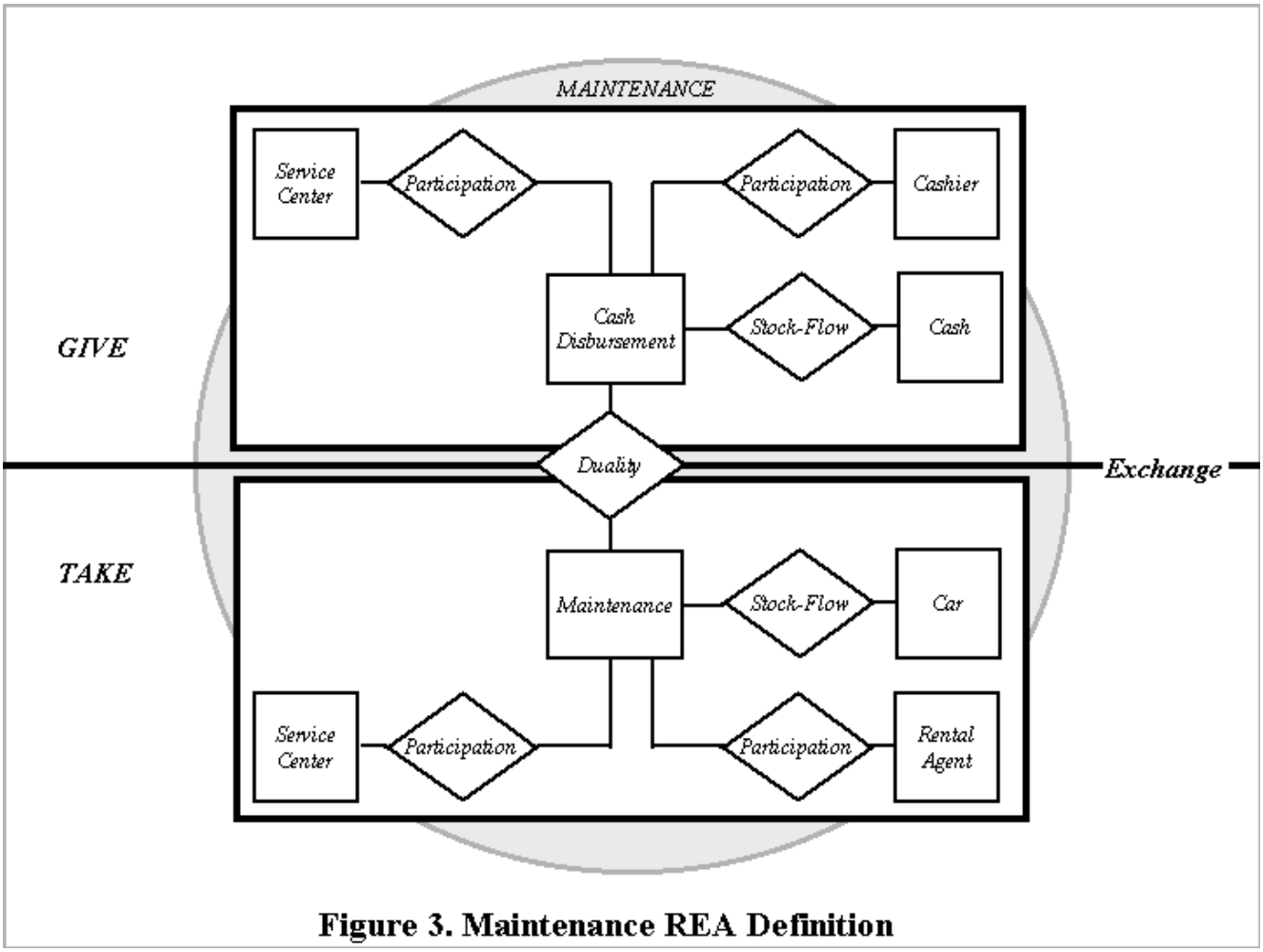

Figure 3. Maintenance REA Definition

Rent-**A**-**C**razy-**C**ar (RACC) is a small company that buys crazy cars (like a pink Cadillac, a madras microbus, a yellow Beetle convertible, a polka-dot hearse, etc.) and rents them out.RACC has a small number of highly specialized employees (agents) who select cars and then rent them. Craziness (or uniqueness) is probably the attribute that customers value most about RACC's products.RACC cars are often rented for weddings, political campaigns, big formal dances, etc.Another attribute customers highly value is that the car they rent works on that "special" day.To realize this objective, the cars are sent out to various automobile service centers at periodic intervals by rental agents for different types of maintenance.

To rent a car, a customer sends a request to RACC.Based on the customer's request, RACC's employees (rental agents) check if the requested type of car is available.When a car is available, an appropriate insurance policy is selected and a contract is prepared.The rental contract is sent to the customer.The customer must pay the rental fee first before he or she can pick up the car.Upon receipt of the payment (check), the rental agent approves the contract and sends a copy to the lot-attendant. The customer goes with his contract to the lot-attendant. The lot-attendant picks up the car and updates the information in the car database.The customer also returns the car to the lot-attendant who updates the product database to close the transaction.

Figure 3 illustrates that we have to give up resources (cash) to acquire a maintained car. The REA object template defines “Maintenance” as an Economic Event resulting in an inflow of Car (or Car Service) from the Service Center (take).Maintenance adds value to the Car resource as the following attribute: “the car should work well on the customer’s special day.”The REA object template in figure 3 also shows “Cash Disbursement” as an Economic Event resulting in an outflow of the Economic Resource “Cash.”

Figure 4 shows a more detailed REA-based description for the “Maintenance” Economic Event.We have added cardinalities using the Batini et al. (1992) notation.Cardinalities are used to express domain-specific rules: that is, business policies applied by the enterprise. The cardinalities for the Car-Maintenance (C-M) Stock-Flow relationship express the following business policies:

[1,1] -- for each maintenance transaction (Economic Event) exactly one car (Economic Resource) is recorded;

[0,n] -- not all cars (Economic Resource) have necessarily been involved in a maintenance transaction (Economic Event) but the same car can be involved in many maintenance transactions.

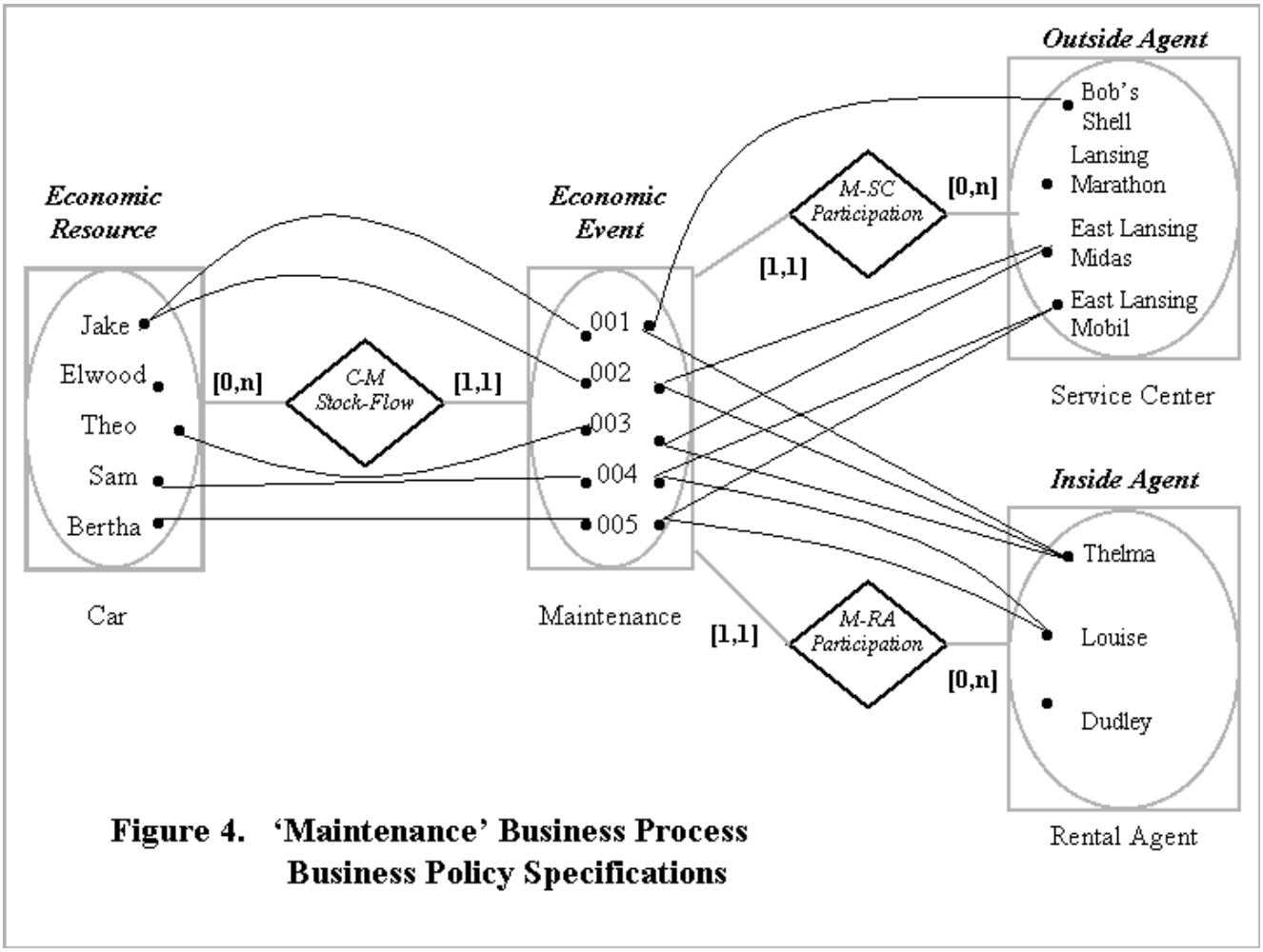

Figure 4. 'Maintenance' Business Process Business Policy Specifications

The other cardinalities in figure 4 express business policies applied by RACC as well.For example, the same Inside Agent can be responsible for many maintenance transactions. Case in point, Thelma is responsible for a set of maintenance transactions: {001,002,003}.RACC often relies on the same Service Center (i.e. the same Outside Agent can be involved in many different transactions).For example, East Lansing Midas has done the maintenance transactions {004,005}.When combined with REA-structures, cardinalities are a powerful tool for the modeling of accounting phenomena. Readers may consult McCarthy (1982),Hollander et al. (2000), Hall (2001) and Romney and Steinbart (2000) for an in-depth discussion of the use of semantic constructs such as cardinalities for modeling accounting phenomena.

Figure 5 shows a partial data model (omitting duality cash transactions) for RACC's significant business activities.The diagram in figure 5 integrates REA-templates for three Economic Events:acquisition of cars [Purchase], maintenance of cars [Maintenance] and the actual renting of cars [Rental Contract].The integrated model precisely describes the acquisition (Purchase, Maintenance) of car services and the use of these car services (Rental Contract).The additional specification of inside agents and outside agents results in an information architecture that supports responsibility and control procedures, planning procedures, and evaluation procedures.

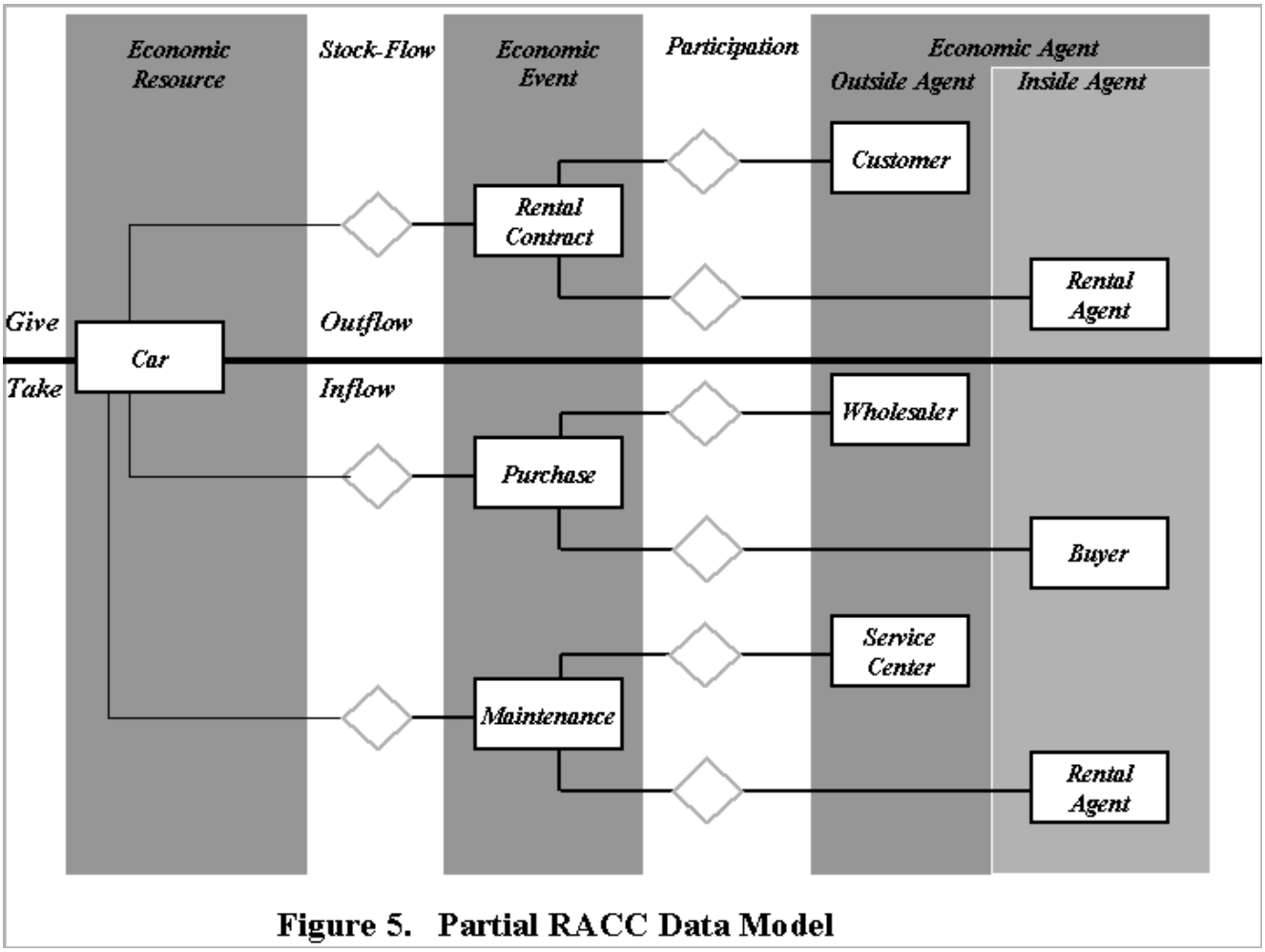

Figure 5. Partial RACC Data Model

In figure 6, we reinstate the cash duality transactions for the processes mentioned in the paragraph above to derive a value chain.The REA logic here models how resources are acquired and consumed in a purposeful manner throughout the enterprise: "*Taken as a whole, duality relationships are the glue that binds a firm's separate economic events together into rational economic processes, while stock-flow relationships weave these processes together into an* ***enterprise value chain"*** (Geerts and McCarthy 97a, p.98). In the next section of the paper, we will focus on the use of this resource acquisition/-consumption network for business process modeling and reengineering.

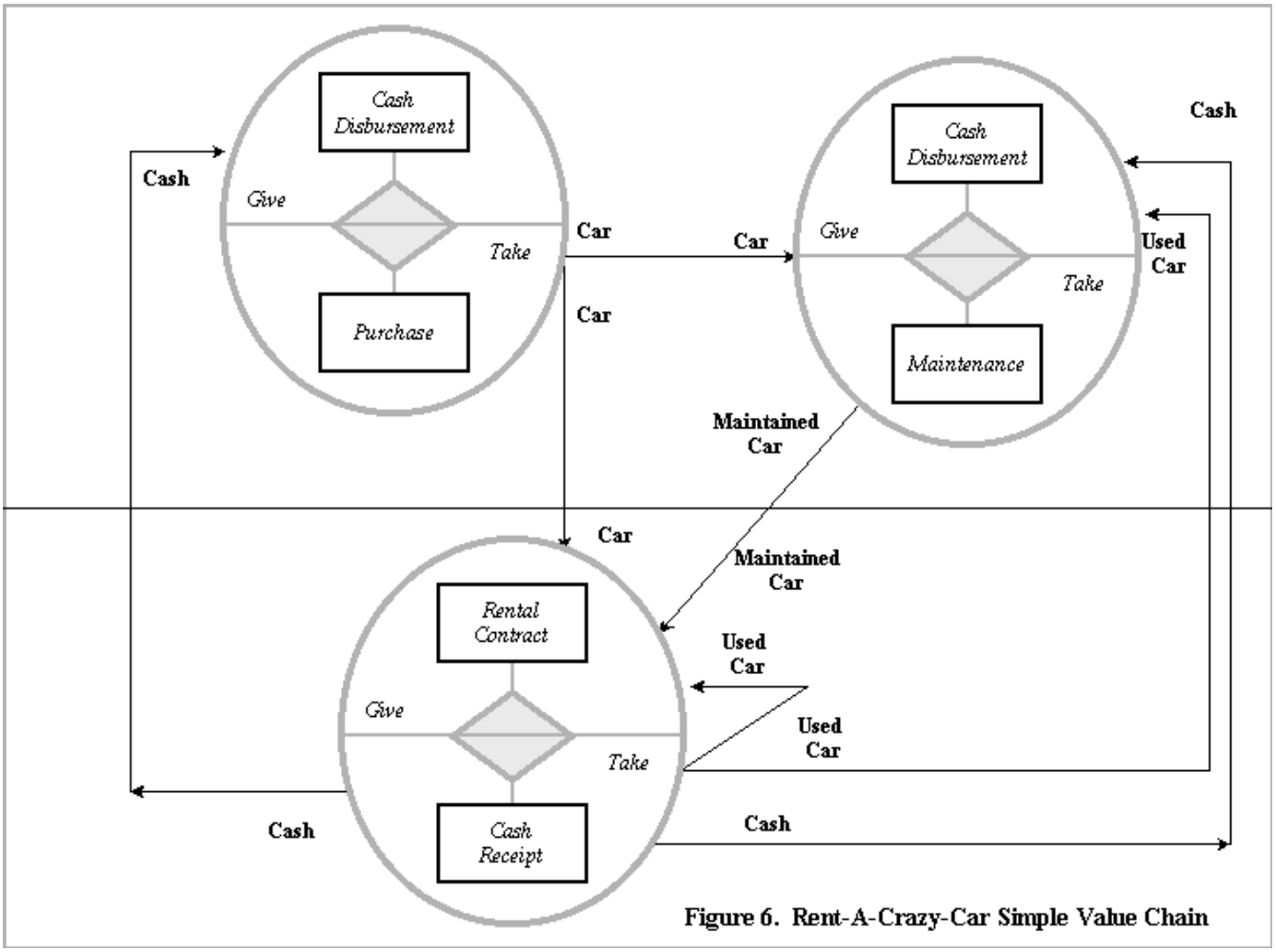

Figure 6. Rent-A-Crazy-Car Simple Value Chain

3. AN REA APPROACH TO BUSINESS PROCESS MODELING

The main goal of business process modeling and reengineering is to manage the integrated organization instead of the stovepiped organization.Integrated process management is possible only when the business processes are known and understood, something that requires a cross-functional synthesis of economic events and workflow tasks into business processes.There are methods available in the literature for accomplishing this.For example, Harrington (1991) discusses a set of graphical techniques to model business processes.Additionally, some object-oriented tools such as Object Life Cycles (Shlaer and Mellor, 1992) or Use-Cases (Jacobson, 1992; Jacobson et al., 1995) are also appropriate.However, as we have seen with our RACC examples, REA logic and organization also seems well suited to this unification process, and we discuss its more detailed use as such in the paragraphs that follow.

3.1 REA events and the need for task-level specification

As we saw in figure 1, business processes can be decomposed into subprocesses multiple times before an enterprise modeler finds the level at which it is appropriate to explode the process into a full set of matched REA patterns.For example, RACC's acquisition process can be decomposed

into vendor selection, order preparation, receiving goods, and vendor payment. Yourdon et al. (1995, p. 137) warn analysts to be "*careful when partitioning a process along the time axis, that each subprocess has a reasonable scope from beginning to end.Consider whether it delivers something of value to the customer.Otherwise there is a danger of falling back into a function-oriented breakdown of processes …"* Again, a working heuristic that gives both an approximate start for object tracking and an approximate finish to process decomposition is to stop at the level at which decision makers need information to plan, design, monitor, and evaluate economic activities.

At the leaf level in a process decomposition of a value chain, an REA template like figure 2 appears.However, a full-REA process model[1] is often more complex than this diagram illustrates because the full set of resource inputs and outputs entails more detail.For example, the revenue process at the bottom of figure 6 shows "Car" (of various types) as input with "Cash" and "Used Car" as outputs, and it is certainly true that in an entrepreneurial logic sense, these are the major process components.That is, we make money by relinquishing the use of our cars for a period of time in turn for which we receive cash from customers.However, it should also be obvious from the company description given earlier that this process consumes other resources such as employee labor.Thus a full-REA model of the revenue process ought to include a "Labor Consumption" event as well as the "Rental Contract" and "Cash Receipt" events.However, full-REA modeling as we describe it is often not technologically attainable, nor is it often decision useful.It is for these cases that we propose an additional level of REA decomposition which we call the ***Task Level***.We illustrate the need for these at the bottom of figure 1, and we describe the rationale for delineating tasks in the section that follows.

3.2 Criteria for Differentiating Between Economic Events and Tasks

In theory, all occurrences in time that consume resources are actually Economic Events in a full-REA world.However, as mentioned above, we sometimes will find it convenient to classify some occurrences as tasks. The criteria for differentiating between an *Economic Event* and a *Task* are highly heuristic and situation-specific in their application.However, the two critical factors in our opinion are these: (1) whether or not an occurrence in time can be paired logically and (somewhat) immediately with an acquisition (increment) event that produces an identifiable and representable resource and (2) whether the specific representation of that occurrence in time is at a level needed to plan, design, monitor, and evaluate.The judgments needed to apply these criteria are explained below with the assistance of the process illustration in figure 7.

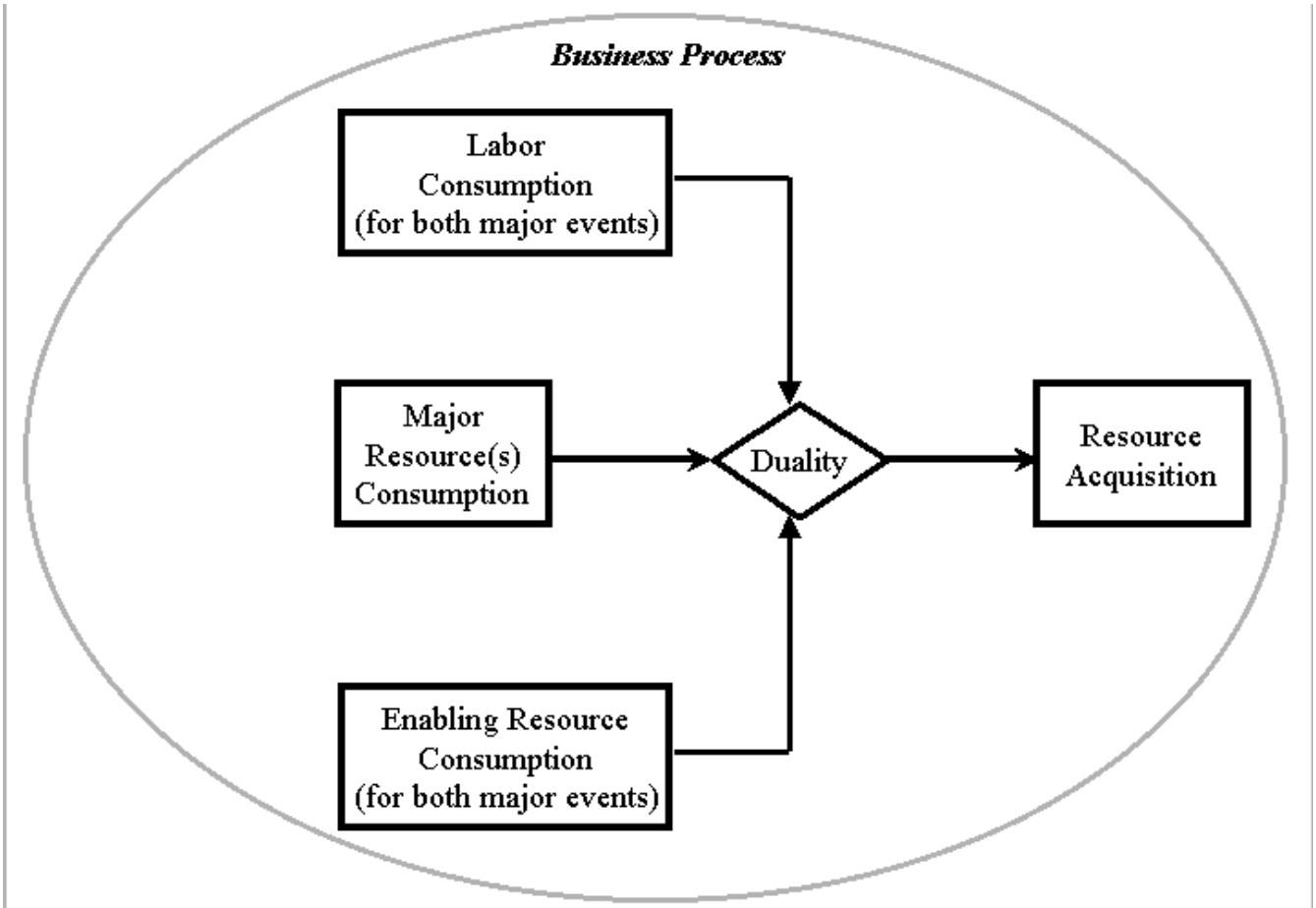

Figure 7. Augmented Consumption-Acquisition Pattern

A process in the entrepreneurial logic sense is an exchange that brings the entrepreneur one step closer to the goal of providing a full value portfolio to the final customer.In this sense, business processes (as illustrated in figure 7) usually have a defining major resource consumption-acquisition pairing.For example, this would be *sale -- cash-receipt* in the revenue process and *cash-disbursement -- purchase* in the materials acquisition process.Almost inevitably these economic conversions will incur other costs as well that enable the exchange.These additional items can be considered transactioncosts in an exchange.For example, the transaction costs for the acquisition and revenue processes mentioned above would include the employee labor use involved in the purchases, sales, cash receipts, and cash disbursements.In addition to labor, processes often entail costs for other resources.Good examples would be car or computer use for a salesperson.We propose that many transaction costs be modeled as tasks instead of as events with the result that their specification is done at the documentation or workflow level as opposed to the data model level.

Representing an occurrence at the task level means rolling its data representation into one of the two major events in the process.For example, the labor use involved in the revenue process wouldn't be represented as a separate consumption event but as an attribute of sale or of cash-receipt.As documentation of the workflow needed to effect a sale, we could specify its tasks (such

as telephone the customer, write down the order, check credit, etc.).This is what we show at the bottom of figure 1 in the task box.

Rolling the representation of labor and enabling-resource consumption events into the major give-take events of a process (that is, making them tasks instead of events) makes sense when some of these conditions apply:

1. notation of the task's completion is clearly immaterial in an information-provision sense (that is, it isn't needed for managerialplanning, designing, monitoring, or evaluating);
2. the task completion process is highly technology-dependent or technology-volatile (that is, both its need and its substance can change with technology);
3. the task doesn't affect an identifiable acquired resource (of value to the customer) whose representation can be materialized until after the completion of all process tasks; and
4. the set of tasks needed to accomplish a consumption event is congruent (that is, its cardinality pattern is (1,1) (1,1)) with the major increment or decrement events of a process.

In an REA setting, using these heuristics is bound to prove difficult at the margin, but we think their application will prove tremendously useful.Separation of resource-consuming occurrences in time into the separate specification groups of economic events and tasks will provide a sound basis for process reengineering.Economic Events are fundamental parts of entrepreneur logic, and they should be reengineered with great reluctance.Tasks on the other hand are less intrinsic to the value-creation process of the firm.As such, they are constant candidates for reevaluation and reengineering.

3.3 Event and Task specification for RACC

Following the discussion above, we can view the revenue process for RACC as a collection of occurrences in time.Examples would be accept customer contact, assess customer needs, check car file & choose the car, assess insurance policy needs and choose, prepare the contract, collect the money, etc.Each of these could be typed as events and exploded as REA templates.However, if we apply the heuristics above, some of them become specified as "the set of tasks needed to accomplish a rental contract and a cash receipt for that contract."As illustrated in figure 8, many of them are now decomposed to the task level (shown as a fishbone diagram at the bottom of the page).Actually, the bottom part of figure 8 illustrates all of the cross-functional tasks needed to accomplish the Revenue Process of RACC.The REA object patterns are used only for the major decrement-increment pairing of *rental contract -- cash-receipt*.

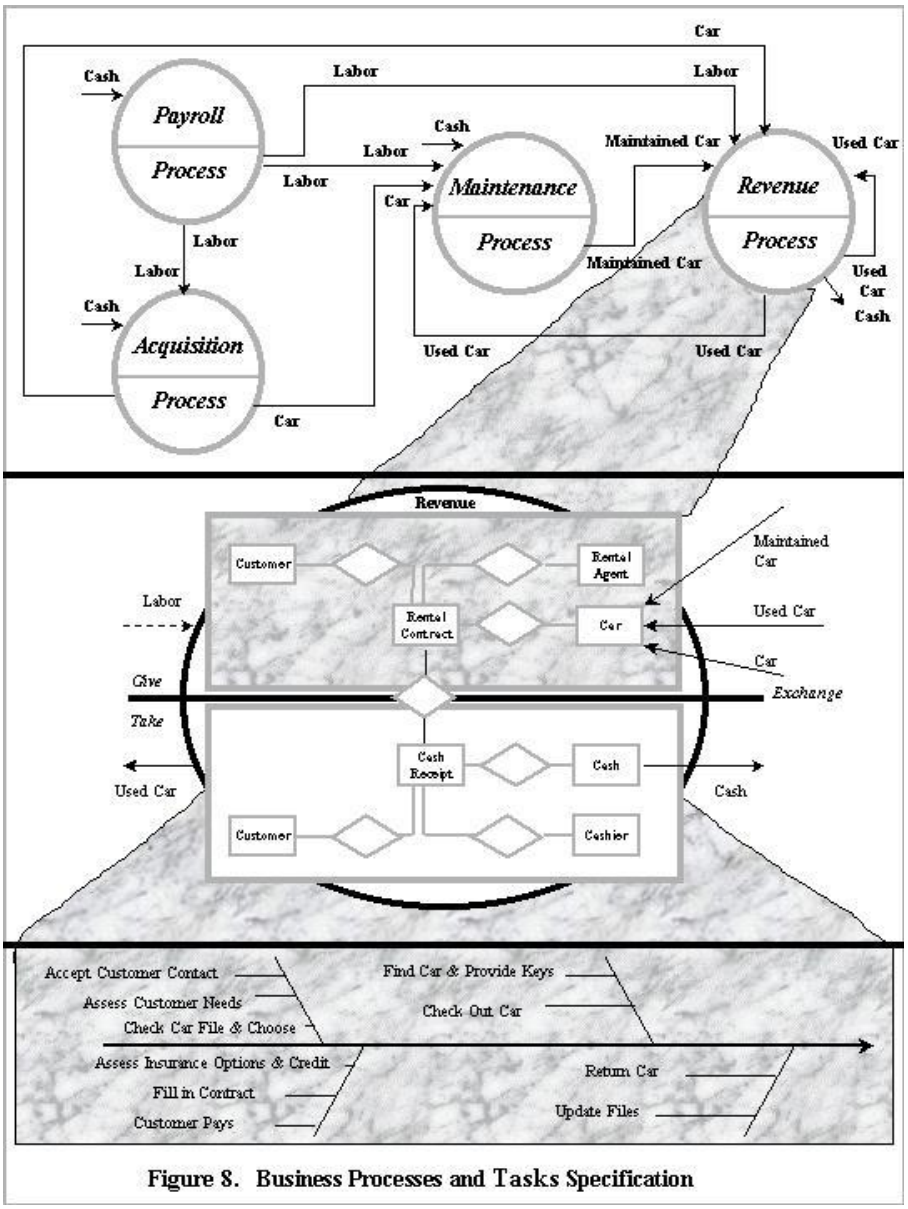

Figure 8. Business Processes and Tasks Specification

Figure 9 illustrates the use of system flowcharts (Hall 2001; Romney and Steinbart 2000; Hollander et al. 2000) to model the cross-functional tasks performed by RACC in support of its revenue cycle.We assume that only two functional departments are involved here: **Rental-Agent** and**Lot-Attendant**.System flowcharts are very helpful in modeling workflow and in documenting communication between functional departments.These flowcharts can also be used to speculate on reengineering possibilities.For example, RACC could decide to let customers "Check and Choose" the cars themselves by giving them Internet access to the Car database.

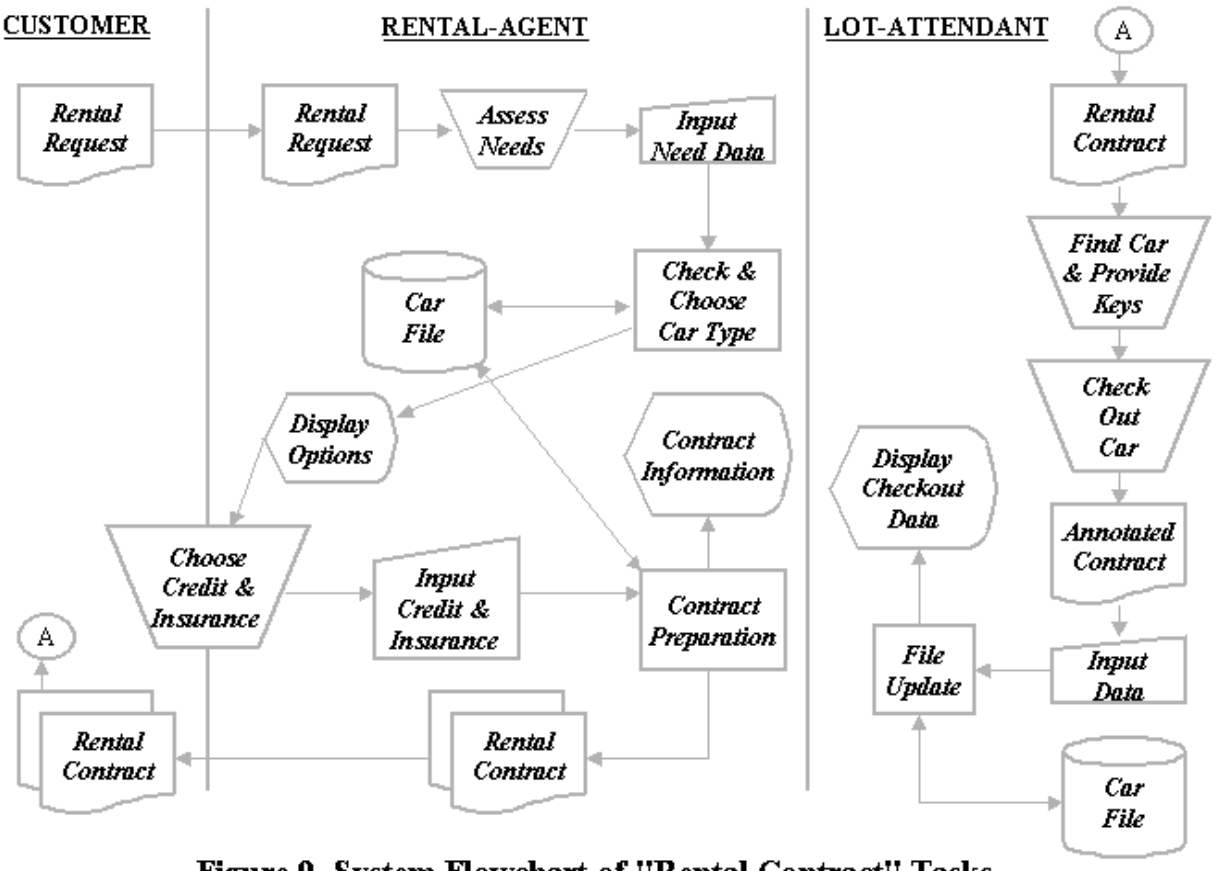

Figure 9. System Flowchart of "Rental Contract" Tasks

Figure 8 summarizes the REA approach to Business Process Modeling. Although not illustrated in figure 8, readers should realize that only some of the business processes will be further decomposed and that only the leaf (bottom-level) business processes in the hierarchy (see figure 1) will be exploded into REA templates.

The REA approach can be used for top-down analysis as well as bottom-up analysis.The "enterprise level" diagram (top of figure 8) integrates the key business processes into a value chain.In addition to the "value-added" processes, the key resource inputs and outputs are clearly specified.Furthermore, it is clearly specified to what extent business processes rely on each other. Figure 8 in the middle shows an REA explosion for a leaf process. The give-take relations describe precisely the value-added transformations for the business process.The invariant entrepreneurial logic of RACC is clearly delineated, and the portion of their enterprise system clearly susceptible to reengineering review on a periodic basis is also clear.

4. CONCLUSIONS AND FURTHER RESEARCH DIRECTIONS

The REA accounting model was introduced in 1982 as: '*A Generalized Framework for Accounting Systems in a Shared Data Environment'* (McCarthy, 1982).A *first* major objective of

the REA framework was to replace functional-driven, isolated file systems by an integrated information structure.Instead of building separate and redundant applications for order processing, shipping, billing, etc., one cross-functional enterprise-wide information structure was to be specified.Database technology has historically enabled the implementation of such an enterprise-wide information structure, and object technology with components promises even better platforms for direct implementation.

A *second* objective of the REA framework was the domain-specific structuring of the enterprise-wide information architecture.Applying the REA template results in a integrated network describing how resources are consumed through value-added processes.This makes REA not only a valuable tool for the actual design of the enterprise information system but also for business process modeling itself.In this paper, we discussed the use of REA object templates to engineer business processes and tasks. We summarize below some of the characteristics of this approach.

- REA can be used for both business process modeling and the actual design of the information system.
- REA supports the domain-specific structuring of the process descriptions, resulting in precise, domain-specific, descriptions of the enterprise value chain.
- The multi-level REA structure allows business process analysis as well as task analysis.That is, we can ask:does a business process add value and can we reengineer the task structure for a certain business process.

This paper is an attempt to use REA for the express purpose of business process modeling.Some important future research efforts that augment the ideas here are listed below.

- System flowcharts alone do not suffice in support of business process management.We need to integrate modeling techniques like workflow management, data flow diagrams, and others discussed in Harrington (1991).
- Object-oriented technology enables a dual role for the REA framework as both a design paradigm and an operational framework.The REA approach to business process modeling should be further integrated with the object-paradigm and most specifically with the full array of specification tools available within UMLJacobson et al. (2000).
- The Event-Task dichotomy introduced here needs to be analyzed and extended in the directions suggested by David (1997).There will certainly be cases where full-REA models will need to be compromised at the representation level to show stand-alone business events (without dualities and resource outflows for example), and heuristics for their specification need to developed further.
- The REA templates used here need to be extended for business process analysis with the additional ontological primitives suggested by Geerts and McCarthy (2000a, 2001) such as commitments and types.Use of these extended REA patterns might possibly move process delineation of tasks and events from more of an art guided by the heuristics given here in section 3.2 to a science governed by basic microeconomic definitions.

5. REFERENCES

Batini, C., Ceri, S. & Navathe,S.B. (1992) *Conceptual Database Design. An Entity Relationship Approach*. Benjamin-Cummings, Redwood City, CA.

Bradley,N. (2000) *The XML Companion.*Addison-Wesley.

Callon, J.D. *Competitive Advantage Through Information Technology*, McGraw Hill, 1996.

David, J. S. (1997) Three "Events" That Define an REA Approach to Systems Analysis, Design, and Implementation. In *Proceedings of the Annual Meeting of the American Accounting Association*, Dallas, TX

Eriksson,H. and M. Penker (2000) *Business Modeling with UML.* John Wiley & Sons.

Fisher,I. (1906) *The Nature of Capital and Income*. MacMillan.

Gal,G. and McCarthy,W.E. (1986) Operations of a Relational Accounting System.*Advances in Accounting*, **3**, 83-112.

Gale,T. and J. Eldred (1996) *Getting Results with the Object-Oriented Enterprise Model*, SIGS Books.

Geerts G.L. and W.E. McCarthy (1994) The economic and strategic structure of REA accounting systems. Paper presented to the 300th Anniversary Program, Martin Luther University, Halle-Wittenberg, Germany.

Geerts G.L. and W.E. McCarthy (1997a) Modeling business enterprises as value-added process hierarchies with resource-event-agent object templates. In: Sutherland J , Patel D, Casanave C, Hollowell G, Miller J, editors. Business object design and implementation. London: Springer-Verlag, pp. 94-113.

Geerts G.L. and W.E. McCarthy (1997b) Using object templates from the REA accounting model to engineer business processes and tasks. Paper presented to the European Accounting Congress, Graz, Austria.

Geerts G.L. and W.E. McCarthy (1999) An accounting object infrastructure for knowledge-based enterprise models.*IEEE Intelligent Systems & Their Application*, 14(4), pp. 89-94.

Geerts G.L. and W.E. McCarthy (2000a) The ontological foundation of REA enterprise information systems. Paper presented to the American Accounting Association Conference, Philadelphia.

Geerts G.L. and W.E. McCarthy (2000b) Augmented intensional reasoning in knowledge-based accounting systems. *Journal of Information Systems*, 14 (2), pp. 127-150 (Fall).

Geerts G.L. and W.E. McCarthy (2001). An ontological analysis of the economic primitives of the extended-REA enterprise information architecture. Paper forthcoming in the *International Journal of Accounting Information Systems*.

Hall, J. (2001) *Accounting Information Systems.* Southwestern.

Hammer,M and Champy,J. (1993), *Reengineering the Corporation.* Harper Business.

Harrington, H.J. (1991), *Business Process Improvement. The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness.* McGraw-Hill.

Hergert,M. and Morris,D. (1989), Accounting Data For Value Chain Analysis. *Strategic Management Journal*, **10**, 175-188.

Hollander, A.S.,Denna E.L. and Cherrington,J.O. (2000) *Accounting, Information Technology and Business Solutions*. Richard D. Irwin, Chicago, IL.

Hoque,R. (2000) *XML for real programmers.* Morgan Kaufmann.

Jacobson,I. (1992) *Object-Oriented Software Engineering: A Use Case Driven Approach.*Addison-Wesley, Reading, MA.

Jacobson,I., Jacobson,M. and Jacobson,A. (1995) *The Object Advantage: Business Process Reengineering with Object Technology*, ACM Press, New York.

Jacobson,I., G. Booch and J. Rumbaugh (1999)The Unified Modeling Language Guide, Addison-Wesley.

McCarthy, W.E. (1982) The REA accounting model: A generalized Framework for Accounting Systems in a Shared Data Environment.*The Accounting Review*, July 1982, 554-578.

Porter, M. E. *Competitive Advantage: Creating and Sustaining Superior Performance*.New York, The Free Press, 1985.

Romney,M.B. and Steinbart P.J. (2000) *Accounting Information Systems*. Addison Wesley.

Shlaer,S. and Mellor S.J. (1992) *Object LifeCycles.Modeling the World in State.* Yourdon Press.

Taylor,D.A. (1995), *Business Engineering with Object Technology.* John Wiley and Sons.

Walden,K. and Nerson J-M (1995) *Seamless Object-Oriented Software Architecture*. Prentice Hall.

Yourdon,E., Whitehead,K., Thomann,J., Oppel,K. & Nevermann,P. (1995) *Mainstream Objects. An Analysis and Design Approach for Business.* Yourdon Press.

Yu, S.C. (1976) *The Structure of Accounting Theory*, The University Press of Florida.

---

[1]The two-way participation relationships between Economic Event and Outside Agent and between Economic Event and Inside Agent replace the 3-way control relationship in McCarthy (1982).

[1] The notion of a full-REA model or an epistemologically adequate schema is explained in great detail by Geerts and McCarthy (2000b).Basically, it means that no stock-flow or duality relationships may be compromised in an REA implementation.In other words, the full REA template must always be instantiated; to do otherwise disables the possibilities for pattern-matched inference.