The Ontological Foundation of REA Enterprise Information Systems

November 1999, March 2000, August 2000

Guido L. Geerts
Assistant Professor of Accounting & MIS
The University of Delaware
Newark, DE 19711
EMAIL: geertsg@be.udel.edu ; TEL: 302-831-6413

&

William E. McCarthy
Professor of Accounting
Michigan State University
East Lansing, MI 48864 USA
EMAIL: mccarth4@msu.edu ;TEL: 517-432-2913

Comments welcomed This paper has benefited from workshop comments received at the University of Florida, the University of Delaware, Arizona State University, the University of San Diego, the University of Kansas, Virginia Tech University, the University of South Florida, the University of Minnesota, the University of Wisconsin, and the 2000 AAA National Meeting. The suggestions of Cheryl Dunn and Julie Smith David are especially acknowledged. The ontological constructs and their definitions have also benefited tremendously from critiques and comments of the ebXML Business Process Team, especially Paul Levine (Telecordia), Karsten Riemer (Sun), and Jim Clark (Edifecs). Our most significant acknowledgement goes to Bob Haugen (Logistical Software) whose insightful commentaries and critiques have caused multiple changes to the ontology.

The Ontological Foundation of REA Enterprise Information Systems

Authors and affiliations

Guido Geerts, The University of Delaware

William E. McCarthy, Michigan State University

ABSTRACT: Philosophers have studied ontologies for centuries in their search for a systematic explanation of existence: "What kind of things exist?" Recently, ontologies have emerged as a major research topic in the fields of artificial intelligence and knowledge management where they address the *content* issue: "What kind of things should we represent?" The answer to that question differs with the scope of the ontology. Ontologies that are subject-independent are called *upper-level ontologies,* and they attempt to define concepts that are shared by all domains, such as time and space. *Domain ontologies*, on the other hand, attempt to define the things that are relevant to a **specific** application domain. Both types of ontologies are becoming increasingly important in the era of the Internet where consistent and machine-readable semantic definitions of economic phenomena become the language of e-commerce. In this paper, we propose the conceptual accounting framework of the ***R***esource-***E***vent-***A***gent (REA) model of McCarthy (1982) as an *enterprise domain ontology*, and we build upon the initial ontology work of Geerts and McCarthy (2000) which explored REA with respect to the ontological categorizations of John Sowa (1999). Because of its conceptual modeling heritage, REA already resembles an established ontology in many declarative (categories) and procedural (axioms) respects, and we also propose here to extend formally that framework both (1) *vertically* in terms of entrepreneurial logic (value chains) and workflow detail, and (2) *horizontally* in terms of type and commitment images of enterprise economic phenomena. A strong emphasis throughout the paper is given to the microeconomic foundations of the category definitions.

I. INTRODUCTION

Philosophers have studied ontologies for centuries in their search for a systematic explanation of existence: "What kind of things exist?" Recently, ontologies have emerged as a major research topic in Information Systems where they address the *content* issue: "What kind of things should we represent?" The answer to that question differs with the scope of the ontology. Ontologies that are subject-independent are called *upper-level ontologies,* and they attempt to define concepts that are shared by all domains, such as time and space. Examples of upper-level ontologies are CYC (Lenat and Guha 1990), John Sowa's ontology (Sowa 1999), and the Weber and Wand ontology (Wand and Weber 1990, Weber 1997) based on the work of Mario Bunge (1977,1979). *Domain ontologies*, on the other hand, attempt to define the things that are relevant to a **specific** application domain. Examples of domain-specific ontologies include air campaign planning (Valente et al. 1999), scheduling (Smith and Becker 1997), and medicine (Gangemi et al. 1998). In this paper, we generally explore the area of enterprise ontologies (Fox et al. 1993, Uschold et al. 1998), and we specifically propose a conceptual accounting framework – the ***R***esource-***E***vent-***A***gent (REA) model (McCarthy 1979,1982) -- as an enterprise ontology. Because of its conceptual modeling heritage, REA already resembles an established ontology, and we also propose here to extend that framework both *vertically* in terms of entrepreneurial logic and workflow detail and *horizontally* in terms of physical-abstract characterizations of enterprise economic phenomena. Before we start our exploration we need to address a number of broader issues: "What is an ontology?" "Why study ontologies?" "How to construct an ontology?" and "What is a good ontology?" Answers to these questions help us to frame the research question addressed in this paper.

What is an ontology?

The most widely accepted definition of ontology is the one given by Gruber (1993): "*an explicit specification of conceptualization*." Gruber uses the description given in Genesereth and Nilsson (1987) to further refine the term conceptualization as "*the objects, concepts, and other entities that are assumed to exist in some area of interest and*

the relationships that hold among them." This definition resembles the traditional description of a database conceptual schema; however, it does differ in at least three important ways: objective, scope and content. First, the objective of an ontology is to represent a conceptualization that is *shareable/reusable* and where idiosyncrasies of specific applications are ignored. Second, the scope of an ontology is *all* applications in the domain, not just one. And finally, an ontology contains *knowledge specifications* where the meaning of the structures represented is explicitly specified and constrained and where the rules to infer further knowledge are explicitly defined.

Why study ontologies?

The interest in ontologies has emerged in the context of the current distributed, heterogeneous computing environment -- in particular the Internet -- and in the fast growing interest in component-based software engineering. The general consensus is that ontologies are *able* to improve *communication, sharing* and *reuse* (Ushold and Gruninger 1996). Lack of an explicitly specified conceptualization often results in poor *communication* from people to people, from people to computers, and especially from computers to computers. For example: "What is the meaning of the terms *account*, *business process* and *market*? Interpretation of these terms is different within departments, across departments, across organizations, and across computer systems. An important objective of ontologies is to make the meaning of concepts explicit in order to improve communication. *Interoperability*, the communication between separate computer systems, has been given increased attention with the emergence of distributed and heterogeneous computing environments, and different approaches have been proposed in support of interoperability. In a first approach, knowledge is translated into a common format such as KIF – the Knowledge Interchange Format (Fikes et al. 1991) -- and this intermediate format can then be used for knowledge sharing and reuse. A second approach is the use of an Agent Communication Language (ACL) that is grounded in an ontology. A good example of an ACL is KQML (Finin et al. 1994). Agents use terms that are part of the ontology to communicate with the underlying implementation being irrelevant.

Lack of *reusability* has been widely recognized as a major weakness of traditional systems development. Reuse can substantially reduce time and cost of information systems design, implementation and maintenance. In recent years much attention is given to the design of software applications where parts of existing applications can be reused. These parts might include designs, knowledge structures, or software components. The accomplishment of reusability largely depends on the sharing of a similar conceptualization.

Many uses of ontologies have emerged with important differences in sophistication and objectives (Musen 1992, Ushold and Gruninger 1996). Here, we distinguish among three broad categories of ontology use: (1) as a knowledge dictionary, (2) as a support for conceptual design, and (3) in operational use.

- *A Knowledge Dictionary* explicitly records the meaning of the domain concepts, the relationships between concepts, and the constraints that apply to concepts. The explicitly recorded definitions improve communication, integration and consistency.
- *Conceptual Design Support.* Ontologies offer important guidance for construction of application models in a specific domain. As Ushold and Gruninger (1996) point out, benefits include a better identification of requirements and increased reliability.
- For *Operational Ontologies,* the concepts, relationships between concepts and constraints are explicitly recorded, and these then become part of the applications themselves (Guarino, 1998). The explicit recording of the ontology as knowledge specifications enables their use for inference.

These uses are neither exhaustive nor exclusive. For example, conceptual design support benefits from the existence of a knowledge dictionary, and an excellent example of operational use of an ontology is the automated support of conceptual design.

How to construct an ontology?

Recognition of the complexity of ontological engineering has resulted in increased research on methodologies for *ontology construction*. Gomez-Perez (1998), for example distinguishes the following phases in ontology construction: knowledge acquisition, requirements specification, conceptualization, implementation, evaluation and documentation. In this paper, we primarily focus on the acquisition, specification and conceptualization of the REA ontology. We believe that at least three different approaches exist to ontology development, with each approach having its own merits: (1) the pragmatic approach, (2) the theoretic approach, and (3) the empirical approach. *The pragmatic approach* defines ontological constructs by solving problems. Currently, this approach is dominant in constructing enterprise ontologies. TOVE, for example, is constructed by addressing specific domain related problems called competency questions (Fox et al. 1993, Gruninger and Fox 1994). *The theoretic approach* derives conceptualizations from existing theories. This is the approach followed in this paper with the enterprise ontology being primarily derived from an existing conceptualization rooted in accounting and economic theory, the Resource-Event-Agent model (McCarthy 1982). No attempts exist to our knowledge to construct ontologies *empirically*. However, a large body of research exists that uses empirical methods to test the ontological theories embodied in human cognition (Rosh et al. 1976, Smith and Mark 1999). These same methods could be applied to ontology building for information systems.

Irrespective of the approach chosen for upfront development, the later phases of ontology construction have to be accomplished before the work is complete. For REA, this implementation-oriented work is the subject of our future research work.

What is a good ontology?

To our knowledge, no framework is available yet to determine the goodness of an ontology. However, many criteria have been proposed. A widely accepted set of criteria is the one discussed in Gruber (1993): clarity, coherence, extendibility, minimal encoding bias, and minimal ontological commitment. First and foremost, ontologies should emphasize the characteristics of *clarity* and *coherence*. Clarity implies that definitions are context independent, and coherence requires consistency of the definitions. The three

other characteristics need a more detailed explanation. The usefulness of an ontology highly depends on its *extendibility***.** Extendibility implies that new concepts can easily be accommodated without any changes to the ontological foundations, for the latter would result in enormous changes in the existing applications. Extendibility is especially important in dynamic environments such as business. An *encoding bias* results when representation choices are made purely for the convenience of notation or implementation. This results in overload and complexity because the representation-specific knowledge and the domain knowledge must be disentangled. Gruber (1993) describes minimal ontological commitment as follows: "*An ontology should make as few claims as possible about the world being modeled, allowing the parties committed to the ontology freedom to specialize and instantiate the ontology as needed.*"

The objective of this paper is to *engineer* the ontological foundation of REA Enterprise Information Systems. This foundation should consist of definitions of the application-independent economic phenomena on top of which enterprise information systems are built. As pointed out by Guarino (1998), the closer we get to defining this reality correctly, the more this knowledge can be reused for *different* tasks. Core economic phenomena included in the REA ontology are exchanges, resource-agent dependencies, resource dependencies, agent dependencies and commitments. Many of these definitions follow from the original REA work, but others are extensions whose usefulness has become apparent in REA development work. We explore each of these phenomena in the second part of the paper. The third part of the paper extends the foundation of the REA ontology vertically into a three-layer framework with support of enterprise phenomena at different levels of granularity: entrepreneur script, process, and task. We also extend the REA ontology horizontally by defining type images for all these phenomena as well as relationships between these type images. Finally, we discuss a number of applications for the REA ontology: corporate memories, conceptual design support and intensional reasoning. We end with some conclusions and further research directions.

II. THE ONTOLOGICAL FOUNDATION OF REA ENTERPRISE INFORMATION SYSTEMS.

The objective of an enterprise ontology is the conceptualization of the common economic phenomena of a business enterprise unaffected by application-specific demands. Sowa (1999) separates concepts to be represented in an ontology into two main categories: physical objects and abstractions[1]. Physical objects describe actual phenomena, while abstractions are information structures that are used to characterize the corresponding physical categories. We follow a similar approach for the REA ontology. The *operational infrastructure* conceptualizes the actual economic phenomena, both current and future. The *knowledge infrastructure* conceptualizes the abstract phenomena that characterize the actual economic phenomena.

Operational Infrastructure

Exchange

The Resource-Event-Agent (REA) framework in McCarthy (1979,1982) is a stereotypical representation of an exchange. The upper part of Figure 1 shows the REA exchange pattern expressed as objects and relationships with ***U***nified ***M***odeling ***L***anguage (UML) notation (Booch et al. 1999). The lower part of Figure 1 illustrates an exchange between finished goods and cash in terms of the REA exchange template.

Similar to many other economists Ijiri (1967, p. 80) considers exchange a core economic phenomenon: "*In a sense, the economic activities of an entity are a sequence of exchanges of resources – the process of giving up some resources to obtain others. Therefore, we have to not only keep track of increases and decreases in the resources that are under the control of the entity but also identify and record which resources were exchanged for which others.*" The REA template captures three intrinsic aspects of exchanges: the requited *events*, the *resources* that are subject of the exchanges, and the participating *agents*. Next, we discuss these three aspects and the REA primitives used to describe them.

[1] Sowa actually refactors this *abstract-concrete* partition into two additional classes of distinctive categories: (a) *continuant-occurrent* and (b) *firstness-secondness-thirdness*. He also traces the philosophical origins and reasoning behind these categorical distinctions. His resultant 2x2x3 factoring

The mirror-image nature of exchanges is represented by the ***duality*** relationship between an inflow Economic Event and an outflow Economic Event. We differentiate between two different types of exchanges -- transfers and transformations (Fisher 1906, Black and Black 1929 -- which leads to two different types of duality relationships: *transformation duality* and *transfer duality*. Transformations create value through changes in form or substance. For transfers, value is created in a market transaction with outside parties. Figure 1 illustrates a transfer.

Stock-flow relationships describe the connection between Economic Resources and Economic Events. Figure 1 differentiates among five different types of stock-flow relationships: *use, consumption, give, take* and *production*. An economic event results in either an inflow or an outflow of resources. Inflows and outflows are further specialized depending on the nature of the duality relationship. For an exchange relationship we *give* up a resource (finished good) to *take* another resource (cash). During a transformation we either *use* or *consume* a resource to *produce* another resource. When resources are used, they often completely disappear in the transformation process and lose their form so as to be unrecognizable. When resources are consumed, they are decremented in chunks that leave the original form discernible (Black and Black 1929, p. 30). It is important to note that the same resource can participate in many different types of stock-flow relationships. For example, a machine is first acquired (take), then employed in production (consumed), and finally sold (give).

The ***participation*** relationship describes the agents involved in an Economic Event. Inside and outside are two different subtypes of this relationship representing the two roles of Agents in the participation relationship. The same agent (person) can be an inside agent (employee) for one event and an outside agent (customer) for another event. We consider accountability as a specific subtype of the inside relationship. An accountability relationship records the (inside) agent responsible for the event.

Exchanges do appear in forms different from the template illustrated in Figure 1. A common example is the situation where the dual events are conceptually congruent.

---

gives 12 top-level ontological categories as displayed in Sowa (1999, chap. 2). An initial analysis of REA primitives in this light is given in Geerts and McCarthy (2000).

Congruency occurs when both events happen simultaneously in *time* and *space*. An exchange with congruent economic events is named a *congruent exchange*. Congruent exchanges have their own intrinsic characteristics and should therefore be differentiated in the ontology. Figure 2 shows the template for a congruent exchange and applies the template to cash-sales. The duality relationship no longer exists and as such is not represented in Figure 2. However, the duality of the exchange is captured by the stock-flow relationships while the nature of the exchange is represented by the subtypes transfer and transformation.

Association, Linkage and Custody

In Figure 3, we add three relationships (shown in shadow) that are not part of an exchange but which conceptualize dependencies between agents (association), between resources (linkage) and between resources and agents (custody). We discuss these dependencies in more detail next.

An ***association*** relationship describes dependencies between agents. We distinguish between three different types of association relationships: *responsibility*, *assignment*, and *cooperation*. The *responsibility* relationship describes a dependency between two inside agents, and McCarthy (1982, p. 564) defined it as follows: "*Responsibility relationships indicate that 'higher level units control and are accountable for activities of subordinates."* It is important to note that an agent does not have to be person but can instead be a department, division or another organizational unit; thus, the responsibility relationship is the vehicle for describing the existing organizational structure. The *assignment* relationship describes dependencies between internal and external agents like a salesperson being assigned to specific customers or a buyer working with specific vendors. Finally, the *cooperation* relationship describes existing dependencies between external agents such as a customer being a subsidiary of a vendor or a joint venture existing between two vendors.

A ***linkage*** relationship describes dependencies between economic resources. An important type of linkage relationship is the composite or part-whole relationship. A composite relationship defines a resource (whole) as an aggregation of two or more other resources (parts). For example, a hard disk, a floppy drive, a monitor, etc. can be defined

as parts of a computer (whole). For possible further specializations of the composite relationship, see Odell (1998) who differentiates between six different part-whole relationships. Linkage relationships exist that don't fit the part-whole structure (non-aggregational relationships). An example of such a relationship is the description of resources that are used as substitutes for another resource.

A ***custody*** relationship describes the internal agent being responsible for a specific resource like the custody relationship between a warehouse clerk and the items stored in the warehouse.

Commitment

At the end of the original REA paper, McCarthy called for extensions into areas such as commitments (McCarthy, 1982, p.576), and the ontological augmentations needed for this are displayed in Figure 4 where **commitment images** for economic events are proposed. Ijiri (1975,p.130) defines a commitment as an "*agreement to execute an economic event in a well-defined future that will result in either an increase of resources or a decrease of resources.*" Commitments are important economic phenomena, and we use Ijiri's term "*executes*" for the relation between them and the actual economic events that follow them. We model the pair-wise connection of requited commitments in a fashion similar to actual exchanges except we substitute a *reciprocal* relationship between the two commitments where an actual exchange has a *duality* relationship. Because of the importance of reciprocal relationships, we take the additional step of reifying them at a higher level of abstraction as *economic agreements,* and we differentiate between two different types of agreements: *contract and schedule,* the definition of which depends on the ultimate nature of the economic exchange. A *transfer executes* a *contract* while a *transformation executes* a *schedule*. For example, a sale executes a sales order which is part of a contract, and a production job executes a production order which is part of a schedule. Two additional relationships are needed to integrate the commitments with the exchange description: *reserves and partner*. *Reserves* is a special kind of stock-flow relationship that describes the scheduled inflow and outflow of resources. A sales order results in a reservation of the finished goods to be delivered, while a production order results in a scheduled completion of finished goods.

Finally, the *partner* relationship is a special kind of participation relationship that describes the outside agents participating in the commitments. We define the partner relationship as a subtype of the outside relationship.

Axioms

It is clear from our discussion thus far that rules exist that restrict the use of the REA primitives for the conceptualization of economic phenomena. The recognition and explicit definition of these rules or *axioms* is an important part of ontological engineering. Bahrami (1999) defines an axiom as: "*a fundamental truth that always is observed to be valid and for which there is no counterexample or exception.*" Axioms are particularly important when they become part of the operational system, which can then reason with them. Next, we define three axioms that are part of the REA ontology.

- ***Axiom1*** -- At least one inflow event and one outflow event exist for each economic resource; conversely inflow and outflow events must affect identifiable resources.
- ***Axiom2*** **--** All events effecting an outflow must be eventually paired in duality relationships with events effecting an inflow and vice-versa.
- ***Axiom3*** -- Each exchange needs an instance of both the inside and outside subsets.

The first axiom guarantees the modeling of the economic activities of a company as a sequence of exchanges. The example in Figure 1 is incomplete since no inflow event is specified for finished good, and no outflow event is specified for cash. These events will become components of exchanges themselves. The second axiom makes sure the correct configurations of exchanges are enumerated, while the third insures the presence of exchanges between parties with competing economic interests.

Knowledge Infrastructure

Type Images

Abstract concepts are information structures used to describe the intangible components of actual phenomena. This is an important philosophical distinction that traces its lineage back to the Greeks (Sowa 1999). In the REA ontology, ***type images*** are

used to represent the intangible structure of economic phenomena. For the construction of type images we use typification, an abstraction commonly used in data modeling (Smith and Smith (1977), Sakai (1981), Brodie (1981) and Odell (1994)).

Typification captures descriptions that apply to a group of actual phenomena. For example, the definition of a lion as a roaring member of the cat family applies to a large number of actual lions. Also important is that the definition of a lion is preserved when lions no longer exist. In the REA ontology, type-images are used to define abstractions of economic phenomena, and this is a distinction that allows us to construct a knowledge-level infrastructure above the transaction-level components (which constitute an operational infrastructure) that were described in the previous section of this paper. Figure 5 integrates the operational and knowledge infrastructures of the REA ontology.

The knowledge infrastructure in Figure 5 contains four different types of images: Economic Resource Type, Commitment Type, Economic Event Type, and Economic Agent Type. Additionally, there would be a type image for Economic Agreement. Instances of the REA types are concepts that apply to a number of resources, events, agents or commitments. An example of an agent type is skills where each skill applies to a number of employees. Other examples of agent types include market segments and agent rankings (such as preferred customers). An example of a resource type is a product class manufactured by a certain company like a Boeing 747[2]. The following categories of sales are examples of event categories: specialty store sales, mail order sales, and Internet sales. And finally, sales orders (commitment) could be typed as immediate-fills, within-policy-fills, late-fills, and never-fills.

A large number of phenomena exist that can be captured by relationships between type images and objects at the operational level. Figure 5 contains such a *description* relationship between Economic Resource Type and Economic Agent. Examples of this relationship might be "the type of resources an agent can provide" or "the type of resources a specific agent is interested in." Another example of a description relationship

[2] Economic Resources like (especially) inventory have an instance/type definition problem that must be solved in the REA ontology (or in any information system) by appeal to the concept of materiality. Thus, small 4" or 3"nails in a hardware store would be modeled with types at operational level and higher types (like roofing nails or finishing nails) at the knowledge level, while cars in an automobile dealership would be modeled with instances (a car with a given engine#) as represented objects in the operational infrastructure with classes of cars (1975 Corvette) as type-images.

(not modeled in Figure 5) is the type of events with which a specific agent can be involved.

Type Image Relationships

The knowledge infrastructure in Figure 5 also shows a number of connections between type images. These connections resemble the relationships at the operational level that were discussed above. At least three different types of abstractions can be captured by type image relationships: *policies*, *prototypes* and *characterizations*. *Policies* are abstractions that restrict the legal configurations of the actual phenomena. An example of a policy for an *assignment* relationship is illustrated in Figure 6. The policy expresses that only an experienced salesperson can be assigned to a large customer. The example in Figure 6 further illustrates that the policy can be used to validate the actual phenomena. *Prototypes* are different from policies in that they do not define restrictions but blueprints. An example of a prototype is a bill of materials expressed in terms of composition relationships between economic resources. Finally, *characterizations* refer to informative type-image relationships. An example of a characterization is the use of the linkage relationship to describe substitutable resource types.

III. THE THREE-LAYER ARCHITECTURE OF REA ENTERPRISE INFORMATION SYSTEMS

Exchanges are the economic unit of analysis in the REA ontology, and the economic activities of a company can be represented as an assembly of purposeful exchanges. However, economic activities must often be viewed at different levels of granularity. Figure 7 shows a top-down decomposition of an ***enterprise script*** as a series of ***processes*** with each of the processes being further exploded into an exchange specification from which itself is derived a script of low-level ***tasks*** needed to accomplish the exchange. The enterprise model illustrated in this figure is derived from an example taken from the rental car industry (Geerts and McCarthy, 1997b). In the next part of the paper, we discuss tasks, enterprise scripts and processes as they relate to the REA ontology. Then, we discuss how this *vertical* layering relates to the *horizontal* layering between the operational infrastructure and knowledge infrastructure.

Operational Infrastructure

Tasks

A task in the REA ontology is a specific compromise of an exchange for which it is not necessary to represent explicitly its dual nature, either conceptually or computationally. The criteria for differentiating between an exchange and a task are highly heuristic and situation-specific in their application. A critical distinguishing feature is whether an outflow event occurrence can be paired logically and (somewhat) immediately with an inflow event occurrence that produces an identifiable and representable resource. Replacing an exchange by a task makes sense when one of the following conditions applies:

1. Notation of a task's completion is clearly immaterial in an information-provision sense (that is, it isn't needed for managerial planning, control, or evaluation), or
2. The task does not affect an identifiable acquired resource whose representation is materialized immediately; instead, the resource is instantiated only after completion of all process tasks.

Replacing an exchange description by a task description results in a representational compromise. As part of the operational infrastructure, it might be useful to capture some of the structural elements of the REA template as part of the task description, even if the full template is not enforced (for example, some of the resources consumed and some of the agents participating in the task). Some possible examples of tasks are discussed below.

- The details of negotiations between a company and a customer for the establishment of a contract such as the assessments of customer needs for the revenue process in Figure 7. It is extremely difficult to *identify and represent* an acquired resource resulting from the use of these labor resources until the contract is signed.
- Labor decrements such as the provision of the keys and updating the files in the car rental example are often considered immaterial at the individual transaction level.

Tasks can sometimes be traced to specific event components of the exchange description, but at other times, they apply to both the increment and the decrement. For simplicity purposes here, we assume that tasks are directly related to the entire exchange.

The fishbone diagram at the bottom of Figure 7 specifies an ordered sequence of tasks. Such an ordered sequence of tasks is called a *recipe* in the REA ontology and a dependency between two tasks is called an *ordering*. The first ordering in the fishbone diagram of Figure 7 *starts* with the task "assess customer needs" and *ends* with the "check car file" task.

Enterprise Scripts and Processes

In the REA ontology, a *process* is defined as an exchange *and* the tasks needed to execute the exchange. For the revenue example in Figure 7, labor and cars are exchanged for cash from customers, and the different tasks needed to execute this exchange (assess customer needs, check car file & choose, etc.) are modeled as a fishbone diagram. The meaning of the term process is related to the one given in Hammer and Champy (1993, p. 53): "*a collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer.*" An *enterprise script* (Geerts and McCarthy, 1999a) describes the actual configuration of processes within a firm. For the example in Figure 7, the enterprise script comprises four processes: payroll, acquisition, maintenance and revenue.

Knowledge Infrastructure

Process Type

Figure 8 defines a knowledge infrastructure for the three-layer architecture. The operational infrastructure in Figure 8 corresponds with the three-layer architecture in Figure 7; however, it must be read from left to right instead of from top to down.

Instances of the revenue process (operational level) are the actual car rentals that take place on a day-to-day basis. The revenue process is itself an instance of process type (knowledge level). Other instances of process type for the car rental problem in Figure 7 are payroll, acquisition and maintenance. The process type image is a vehicle to define

generic characteristics for each of these processes such as their objectives. These generic characteristics then apply at the class level to all the individual instances of that process. For the example in Figure 8, the generic characteristics of the revenue process (an instance of process type) apply to both RP#1 and RP#2. RP#1 and RP#2 are actual car rentals (instances of the revenue process).

Exchange Type

In the top middle of Figure 8, we define a type image for exchange. Because of space limitations, only part of the exchange template -- an outside relationship between agent and event -- is shown at both the operational and the knowledge level. The complete structure of Exchange Type corresponds with the REA template in Figure 1, and the complete structure for the revenue cycle exchange is represented as the middle layer in Figure 7. Again, generic characteristics defined for an exchange type apply to all the actual instances of that type.

Task Type & Recipe Type

A task type defines generic characteristics for all actual tasks to which the type definition applies. In Figure 8, we show two instances of task type, which could be for example "Assess Customer Needs" and "Check Car File and Choose." The instances of the first task are the actual assessments that take place. An ordering type defines dependencies among task types. Here we specify that the assessment will take place first and the checking of the car file next. Instances of the ordering type are then used to define a recipe type, which is *an ordered sequence of task types to be executed in association with a process type*. The fishbone diagram in Figure 7 represents a recipe type to be executed within the revenue process, and an actual car rental is expected to go through the different tasks in the specified sequence.

It is important to note that different recipe types can be defined for the same process type. These could include for instance: the recipe currently applied, the best practice recipe(s) as embodied in published industry standards or in different types of software, the recipes that have the lower costs but which have had undesirable side effects, etc.

IV. REA ONTOLOGY APPLICATIONS: CORPORATE MEMORIES, CONCEPTUAL DESIGN SUPPORT, AND INTENSIONAL REASONING

Ontologies can be applied in many ways with important differences in complexity and objectives. In this section, we discuss three possible applications for the REA ontology: corporate memories, conceptual design support and intensional reasoning. The applications are similar to the uses of ontologies discussed in section one.

Corporate Memories

An application of knowledge technology for which there is growing interest is corporate memories. Corporate memories or knowledge management systems "*capture a company's accumulated know-how and other knowledge assets and make them available to enhance the efficiency and effectiveness of knowledge-intensive processes"* (Kuhn and Abecker 1998). Corporate memories result in knowledge capitalization which, as Arrow (1999) points out, can result in an extreme form of increasing returns since the same knowledge can be applied over and over again. Ontologies are vital to corporate memories for the reuse and sharing of accumulated knowledge, and readers may consult sources like O'Leary (1998) for more detailed discussions of the factors that drive the need for ontologies in knowledge management. A distinguishing feature of the REA ontology is that it provides specific guidance regarding the types of knowledge to be captured and represented in a corporate memory, such as best practices for processes, detailed descriptions of resources and composite structures between resources, and expertise needed for execution of specific tasks.

Conceptual Design Support

An emerging use of ontologies is as reusable analysis patterns during the requirement analysis phase of system design (Ushold and Gruninger (1996), Smith and Becker (1997) and Johannesson and Wohed (1998)). As an analysis pattern, ontologies specify the objects of interest in the domain and the rules to assemble objects into information structures. In addition, experiences and best practices for the patterns are stored in a knowledge base to be shared and reused. As Ushold and Gruninger (1996)

point out, the benefits of using ontologies for construction of application models include better identification of requirements and increased reliability

An important feature of the REA ontology is the inclusion of domain-specific rules that help to structure economic phenomena. The following two rules, defined as axioms 1 and 2 in the previous section, illustrate the structuring capabilities of the REA ontology:

- The duality relationship must be enforced, a situation that forces analysts and designers to consider explicitly the causal links between events and consequently between resources. For example: Why did we use labor? What resources where consumed to inspect a process? What was the purpose of providing a client with certain non-transaction benefits?
- Both an inflow event and an outflow event must be specified for an economic resource. This rule insures that resources are purposely acquired and that exchanges are combined into an enterprise script.

Geerts and McCarthy (1992) have built an intelligent CASE tool, CREASY, that uses both of these rules to force designers to think of enterprise models as an enterprise script of economic exchanges. In another CASE tool, REACH (Rockwell 1992, Rockwell and McCarthy 1999), three different types of knowledge are used for the integration of different conceptualizations. These three different types of knowledge are first-order principles of the REA model, heuristic guidance of implementation compromises based on object pattern matches, and reconstructive expertise for prototypical models. The first-order principles correspond with the REA primitives. The two other types of knowledge correspond with best practices and experiences for the implementation of REA patterns.

Intensional Reasoning

The most advanced use of ontologies is **operationally** where the ontological specifications are explicitly recorded and where they can consequently be used in automated tasks such as conceptual design, information retrieval and problem solving.

Geerts and McCarthy (1999b) use the REA ontology for reasoning about accounting concepts. They use the language Prolog to record explicitly both data (operational) and a knowledge infrastructure that consisted of a conceptual schema, a set of declarative primitives, and a taxonomy of shareable and reusable accounting concepts. The accounting concepts were defined in terms of REA primitives, and the conceptual schema was an instantiation of an REA enterprise script. Then, they built different types of applications for which the explicitly-recorded knowledge was used. They call their use of the knowledge definitions for problem solving "intensional reasoning," where *intension* refers to the meaning of the concepts used. One of their applications was intelligent information retrieval wherein the CREASY system was able to detect different instances of claims based on a formal definition such as: "*A claim with an outside agent exists where there is a flow of resources with that agent without the full set of corresponding instances of a dual flow.*" Their system recognized the following types of claims in their examples: accounts-receivable, accounts-payable and prepayments. The CREASY system demonstrated that a strong degree of "ontological commitment" to a certain domain theory could be used to enable the construction of an extendible taxonomy of accounting concepts, something which in turn could be shared and reused across a variety of applications.

For other examples of the importance of operational uses of ontologies, readers may consult Gruninger and Fox (1994), Bergamashi et al. (1999) and Jin et al. (1998).

V. CONCLUSIONS AND FURTHER RESEARCH DIRECTIONS

This paper addresses the content issue of enterprise ontologies as it tries to answer the question of "What phenomena should be represented in Enterprise Information Systems?" Our starting point was an existing conceptual accounting framework strongly rooted in economic and accounting theory: The Resource-Event-Agent (REA) model. Our concluding point was an augmented set of REA ontological primitives with the following structure:

The Operational Infrastructure

1. *Exchange* as defined in McCarthy (1982) with the REA pattern and as extended here with *commitment* entities and with *association*, *custody* and *linkage* relationships.
2. The three-level architecture with *enterprise script*, *process*, *recipe*, *ordering* and *task* primitives to describe *exchanges* at different levels of granularity.

The Knowledge Infrastructure

3. *Type images* and *type image relationships* to capture generic information about the actual phenomena and to specify policy level structuring.

The ontological development work begun here is certainly not finished. The structures outlined above are just the beginning of a program of research in this area that we hope to continue for some time as the value to enterprise commerce of good ontologies increases in magnitude. Improvement and development areas in the future include the following:

- First, all of the ontological components defined thus far, and any future ones as well, need to become more grounded in economic theory. To a certain extent, we have relied on our own background in economics and on the economic insights of business scholars like Yuji Ijiri (1975) and Michael Porter (1985). We need to develop this background further, in particular with the type of extensions that are being used in the theory of the firm (Carroll and Teece (1999), Putterman and Kroszner (1996)).

- Second, the enterprise information architectures suggested here at two levels (the operational level and the knowledge level) need to be extended and integrated with other domain-specific ontologies from closely related fields of business like products and product development, supply chain management, and business organization and planning. We also need to integrate components of other enterprise ontologies that fit within the theoretical patterns and parameters of our extended REA models.

- And finally, the ontological engineering aspects of our REA ontology need development. The definitions need to be converted into a formal language such as Ontolingua to discover inconsistencies and to analyze operational use. The robustness of the ontological constructs need be evaluated in as many different contexts as possible and the set of axioms needs to be expanded considerably to include a wide variety of enterprise-related definitions.

REFERENCES

Arrow,K. 1999. "Technical Information and Industrial Structure," in G.R.Carroll and D.J. Teece, eds., *Firms,Markets, and Hierarchies*, Oxford University Press.

Bahrami, A. 1999. *Object-Oriented Systems Development,* Irwin McGraw-Hill.

Bergamashi, S., S. Castano, S. De Capitani di Vimercati, S. Montanari, and M. Vincini. 1998. "An Intelligent Approach to Information Integration, "in N.Guarino, ed., *Formal Ontologies in Information Systems*, IOS Press.

Black,J.D. and A.G. Black. 1929. *Production Organization*. Henry Holt and Company.

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

Brodie,M.L. 1981. "Association: A Database Abstraction for Semantic Modeling," in P.P.Chen, ed*., Entity-Relationship Approach to Information Modeling and Analysis*, ER Institute, pp. 583-603.

Bunge, M. 1977. *Treatise on Basic Philosophy: Ontology I – The Furniture of the World*, Reidel, 1977.

Bunge, M. 1979. *Treatise on Basic Philosophy: Ontology II – A World of Systems*, Reidel, 1979.

Fikes,R., M. Cutkosky, T.R. Gruber, and J.V. Baalen. 1991. *Knowledge Sharing Technology Project Overview*, Knowledge Systems Laboratory, KSL-91-71, Stanford University.

Finin,T., Y.Labrou and J.Mayfield. 1994. "KQML as an Agent Communication Language," *Proceedings of the Third International Conference on Information and Knowledge Management*, Gaithersburg, Maryland, ACM Press, pp. 456-463.

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

Fowler,M. 1997. *Analysis Patterns. Reusable Object Models*, Addison Wesley.

Fox,M.S., J.F.Chionglo and F.G.Fadel. 1993. "A Common Sense Model of the Enterprise," *Proceedings of the 2nd Industrial Engineering Research Conference*, pp. 425-429, Norcross, GA, USA, Institute for Industrial Engineers.

Gangemi,A., D.M.Pisanelli and G. Steve. 1998. "Ontology Integration: Experiences with Medical Terminologies," in N.Guarino, ed., *Formal Ontologies in Information Systems*, IOS Press.

Geerts, G. and W.E. McCarthy. 1992. "The Extended Use of Intensional Reasoning and Epistemologically Adequate Representations in Knowledge-Based Accounting Systems." *Proceedings of the Twelfth International Workshop on Expert Systems and Their Applications*, Avignon, France (June): 321-32.

Geerts, G. and W.E. McCarthy. 1997a. "Modeling Business Enterprises as Value-Added Process Hierarchies with Resource-Event-Agent Object Templates," in Jeff Sutherland and Dilip Patel, eds., *Business Object Design and Implementation*, Springer-Verlag, 1997, pp. 94-113

Geerts, G. 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. and W.E. McCarthy. 1999a. "An Accounting Object Infrastructure For Knowledge-Based Enterprise Models." *IEEE Intelligent Systems & Their Applications* (July August 1999), pp. 89-94.

Geerts, G. and W.E. McCarthy. 1999b. "Augmented Intensional Reasoning in Knowledge-Based Accounting Systems," Forthcoming in *Journal of Information Systems*

Geerts, G. and W.E. McCarthy. 2000. "An Ontological Analysis of the Primitives of the Extended-REA Enterprise Information Architecture," Forthcoming in *The International Journal of Accounting Information Systems*

Genesereth,M.R. and N.J. Nilsson. 1987. *Logical Foundation of Artifical Intelligence*, Morgan Kaufmann, Los Altos, California.

Gomez-Perez, A. 1998. "Knowledge sharing and reuse," in J. Liebowitz, ed., *The Handbook of Applied Expert Systems*, CRC Press.

Guarino,N. 1998. "Formal Ontology and Information Systems," in N. Guarino, ed., *Formal Ontology and Information Systems*, IOS Press.

Gruber, T. 1993. "A Translation Approach to Portable Ontologies," *Knowledge Acquisition*, pp. 199-220.

Gruninger,M. and M.S.Fox. 1994. "The Role Of Competency Questions in Enterprise Engineering," *Proceedings IFIP WG5.7 Workshop on Benchmarking – Theory and Practice*, Trondheim, Norway, 1994.

Hammer,M. and J. Champy. 1993. *Reengineering the Corporation: A Manifesto for Business Revolution*, Harper Business.

Ijiri, Y. 1967. *The Foundations of Accounting Measurement,* Prentice-Hall.

Ijiri,Y. 1975. *Theory of Accounting Measurement,* American Accounting Association.

Jin, Z., D. Bell, F.G. Wilkie and D. Leahy. 1998. "Automatically Acquiring Requirements of Business Information Systems by Reusing Business Ontology," *Proceedings Workshop on Applications of Ontologies and Problem-Solving Methods, 13th European Conference on Artificial Intelligence*, pp78-87

Johannesson,P. and P.Wohed. 1998. "Deontic Specification Patterns – Generalisation and Classification," in N.Guarino, ed., *Formal Ontology in Information Systems*, IOS Press.

Kuhn,O and A.Abecker. 1998. "Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges," in U.M.Borghoff and R.Pareshi, eds., *Information Technology for Knowledge Management*, Springer, pp. 183-206.

Lenat,D.B. and R.V. Guha. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the CYC project,* Addison-Wesley Publishing Company, Inc., Reading, Massachusetts.

McCarthy, W.E. 1979. An Entity-Relationship View Of Accounting Models. *The Accounting Review* (October): 667-86.

McCarthy, W.E. 1982. "The REA Accounting Model: A Generalized Framework for Accounting Systems in A Shared Data Environment." *The Accounting Review* (July), pp. 554-578.

Musen,M.A. 1992. "Dimensions of Knowledge Sharing and Reuse*," Computers and Biomedical Research* 25, pp. 435-467

Odell,J.J. 1998. *Advanced Object-Oriented Analysis & Design Using UML,* Cambridge University Press.

O'Leary,D. 1998. "Using AI in Knowledge Management: Knowledge Bases and Ontologies," *IEEE Intelligent Systems*, May/June, pp. 34-39.

Porter, M.E. 1985. *Competitive Advantage.* The Free Press.

Putterman,L. and R.S. Kroszner. 1996. *The Economic Nature of the Firm.* Cambridge University Press.

Rockwell,S.R. 1992. *The Conceptual Modeling and Automated Use of Reconstructive Accounting Knowledge*, unpublished dissertation, Michigan State University.

Rockwell, S.R. and W.E. McCarthy. 1999. "REACH: Automated Database Design Integrating First-Order Theories, Reconstructive Expertise, And Implementation Heuristics for Accounting Information Systems," forthcoming in *International Journal of Intelligent Systems in Accounting, Management, and Finance.*

Rosh,E., C.B.Mervis, W.D.Gray,D.M. Johnson and P.Boyes-Braem. 1976. "Basic Objects in Natural Categories," *Cognitive Psychology*, 8, pp. 38-439.

Sakai, H. 1981. "A Method for Defining Information Structures and Transactions In Conceptual Schema Design*," Proceedings of the Seventh International Conference on Very Large Data Bases*, pp. 225-34.

Smith,B. and D.Mark. 1999. "Ontology with Human Subjects Testing: An Empirical Investigation of Geographic Categories," *American Journal of Economic and Sociology*, Vol. 58, No. 2, pp. 246-265.

Smith, S.F., and Becker, M.A.. 1997. "An Ontology for Constructing Scheduling Systems", Working Notes of 1997 AAAI Symposium on Ontological Engineering, Stanford, CA, March. AAAI Press.

Smith, J.M. and D.C.P. Smith. 1977. "Database abstractions: Aggregation and Generalization," *ACM Transactions on Database Systems* (June): 105-133.

Sowa, J. 1999. *Knowledge Representation: Logical, Philosophical, and Computational Foundations,* Brooks/Cole Publishing, Pacific Grove, CA.

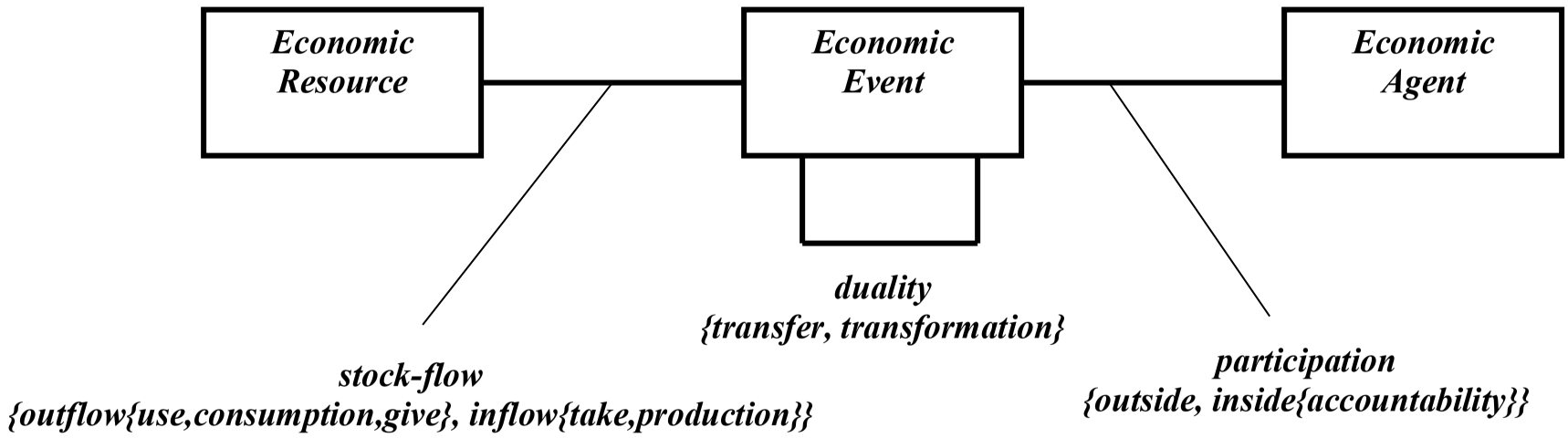
Uschold,M., M.King and S.Moralee. 1998. "The Enterprise Ontology," *The Knowledge Engineering Review*, 13(1), pp.31-89.

Ushold,M. and M.Gruninger. 1996. "Ontologies: Principles, Methods and Applications," *The Knowledge Engineering Review*, 11(2), pp. 93-136.

Valente,A., T.Russ, R.MacGregor and W.Swartout. 1999. ″Building and (Re)Using an Ontology of Air Campaign Planning″ *IEEE Intelligent Systems* , Jan-Feb, pp. 27-36.

Wand, Y. and R. Weber. 1990. "An Ontological Model of an Information System," *IEEE Transactions on Software Engineering*, November, pp. 1282-92.

Weber, R. 1997. *Ontological Foundations of Information Systems*, Queensland, Australia, Coopers & Lybrand.



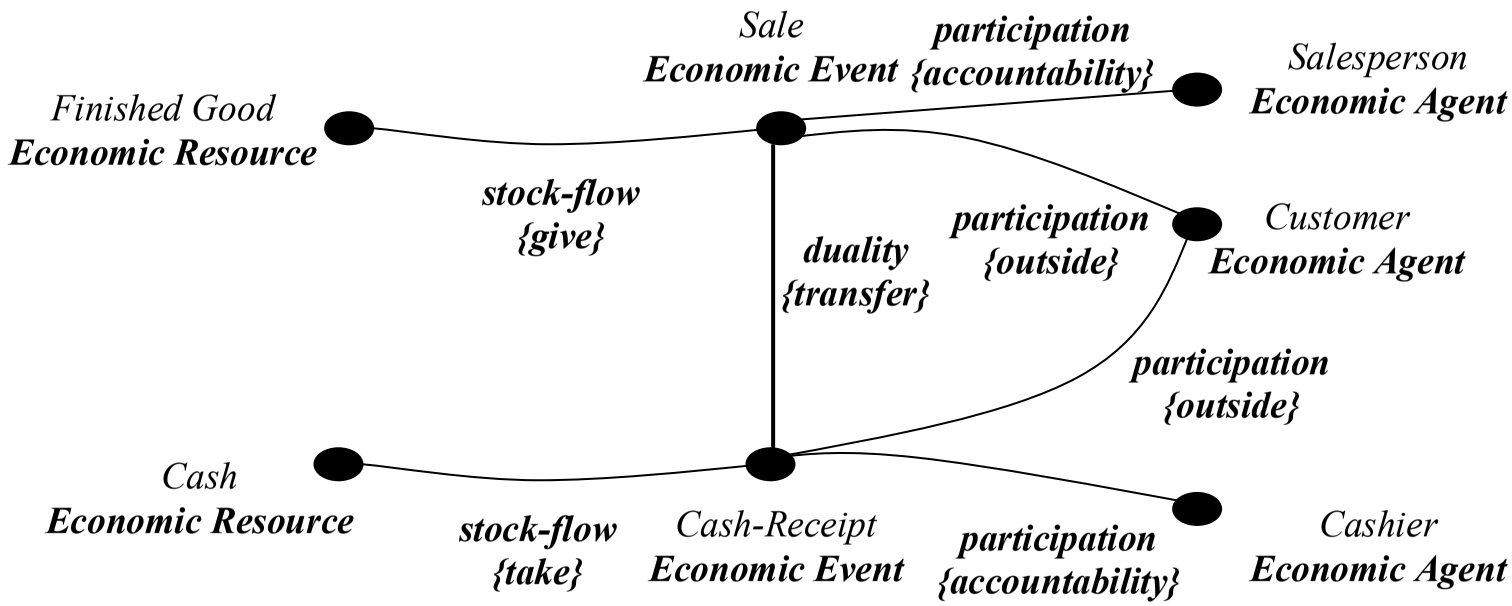
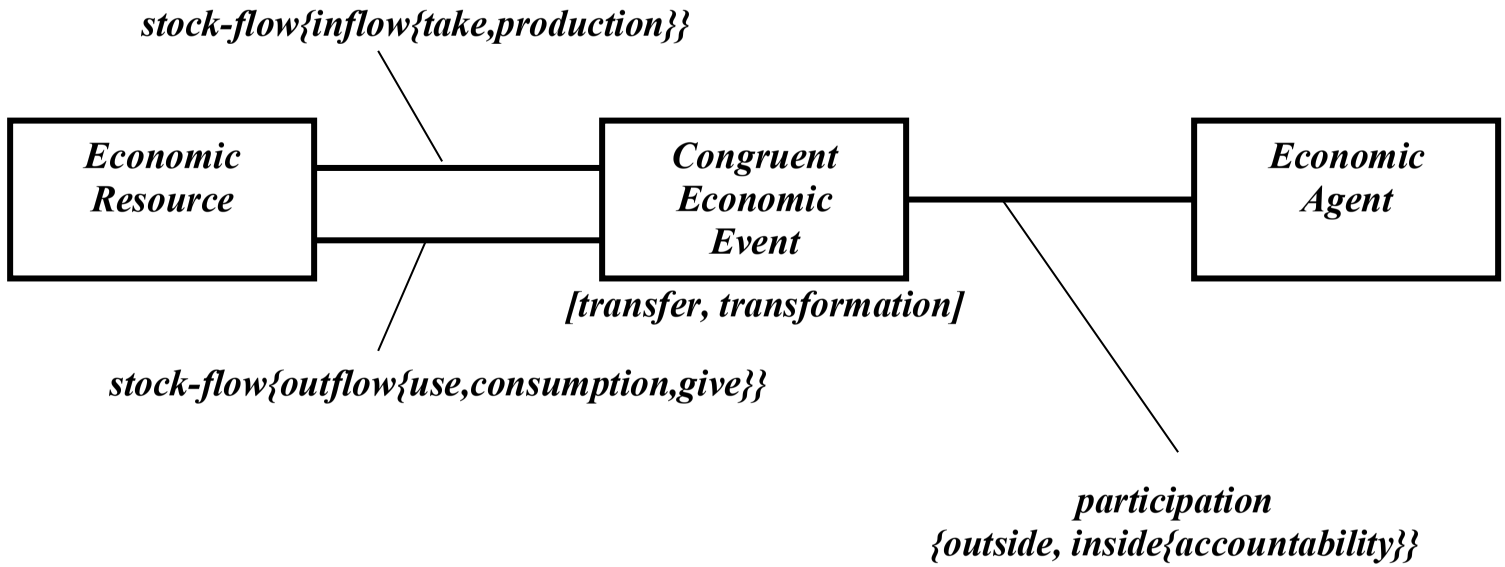


Figure 1 -- Exchange



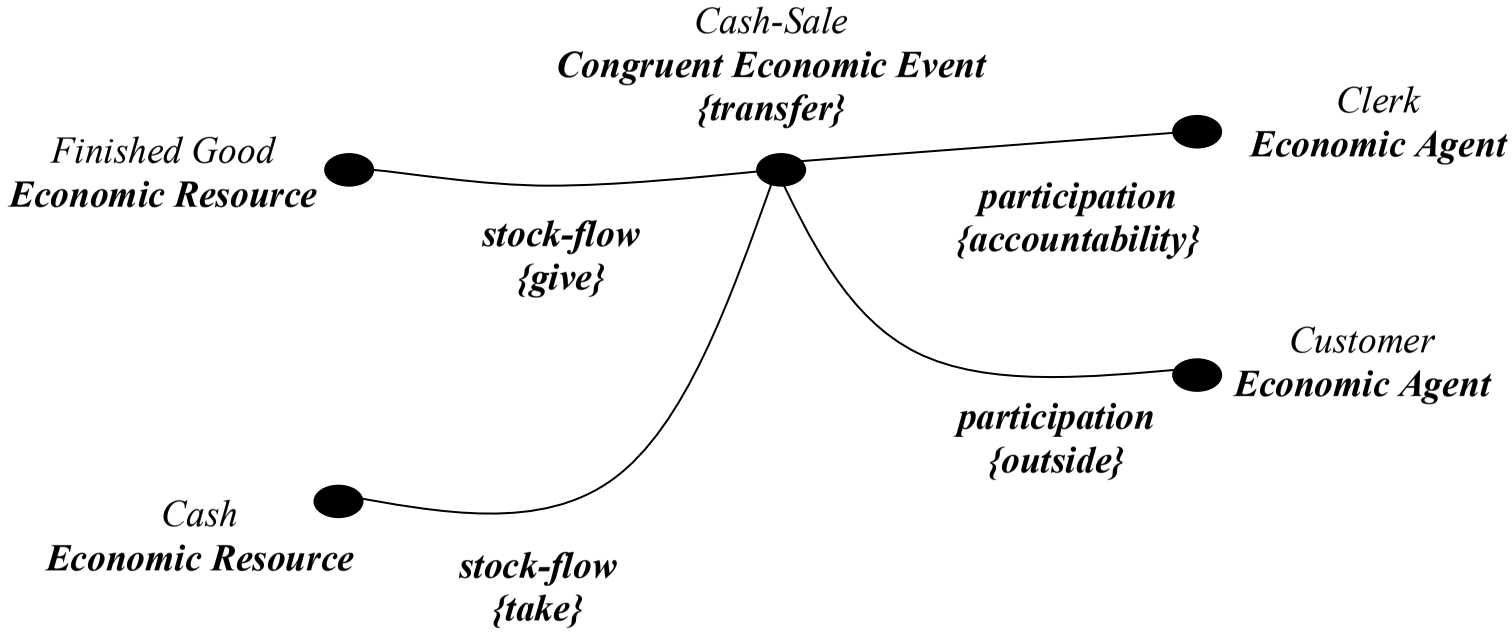


Figure 2 -- Congruent Economic Exchange

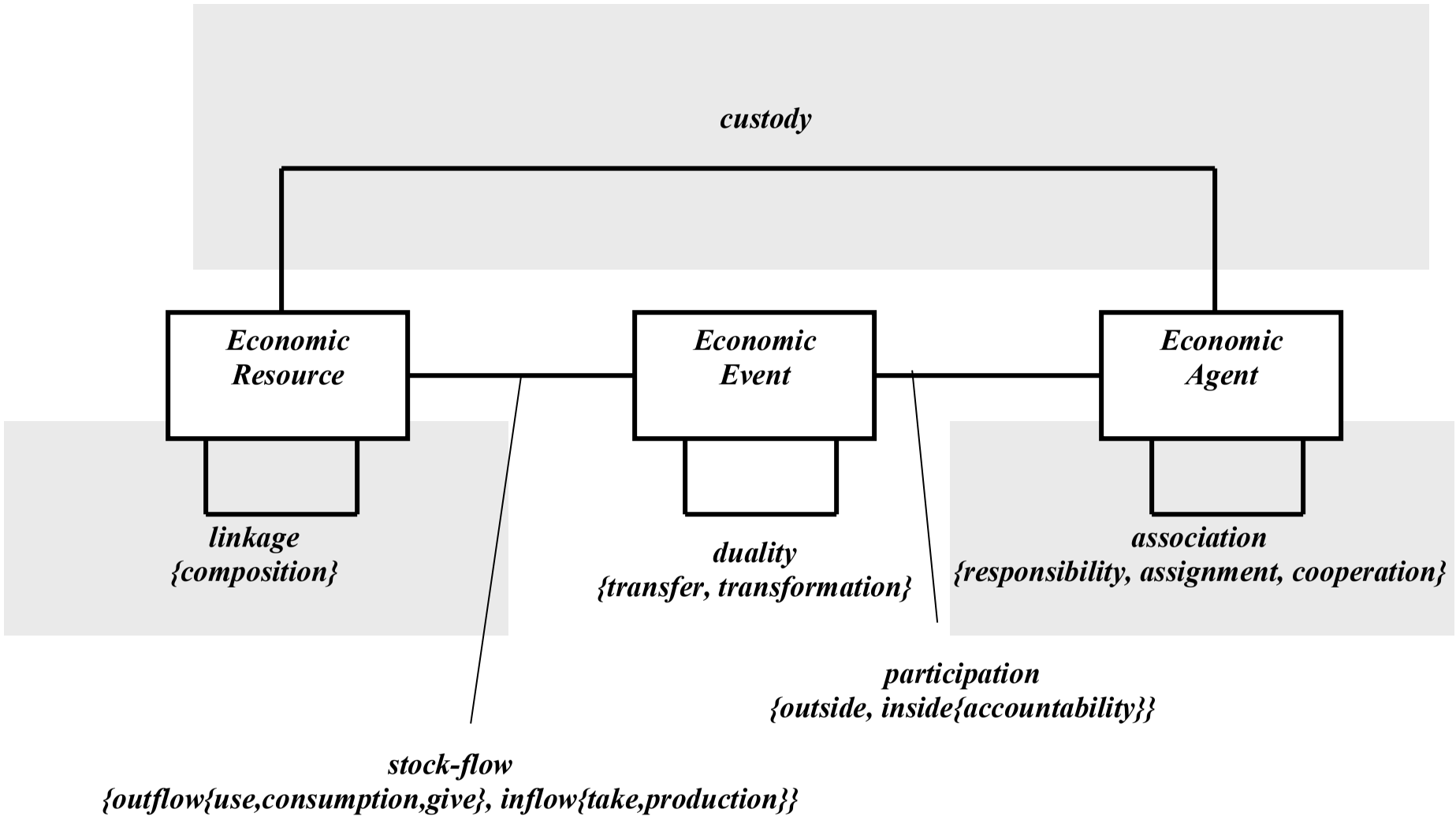


Figure 3 -- Association, Linkage and Custody

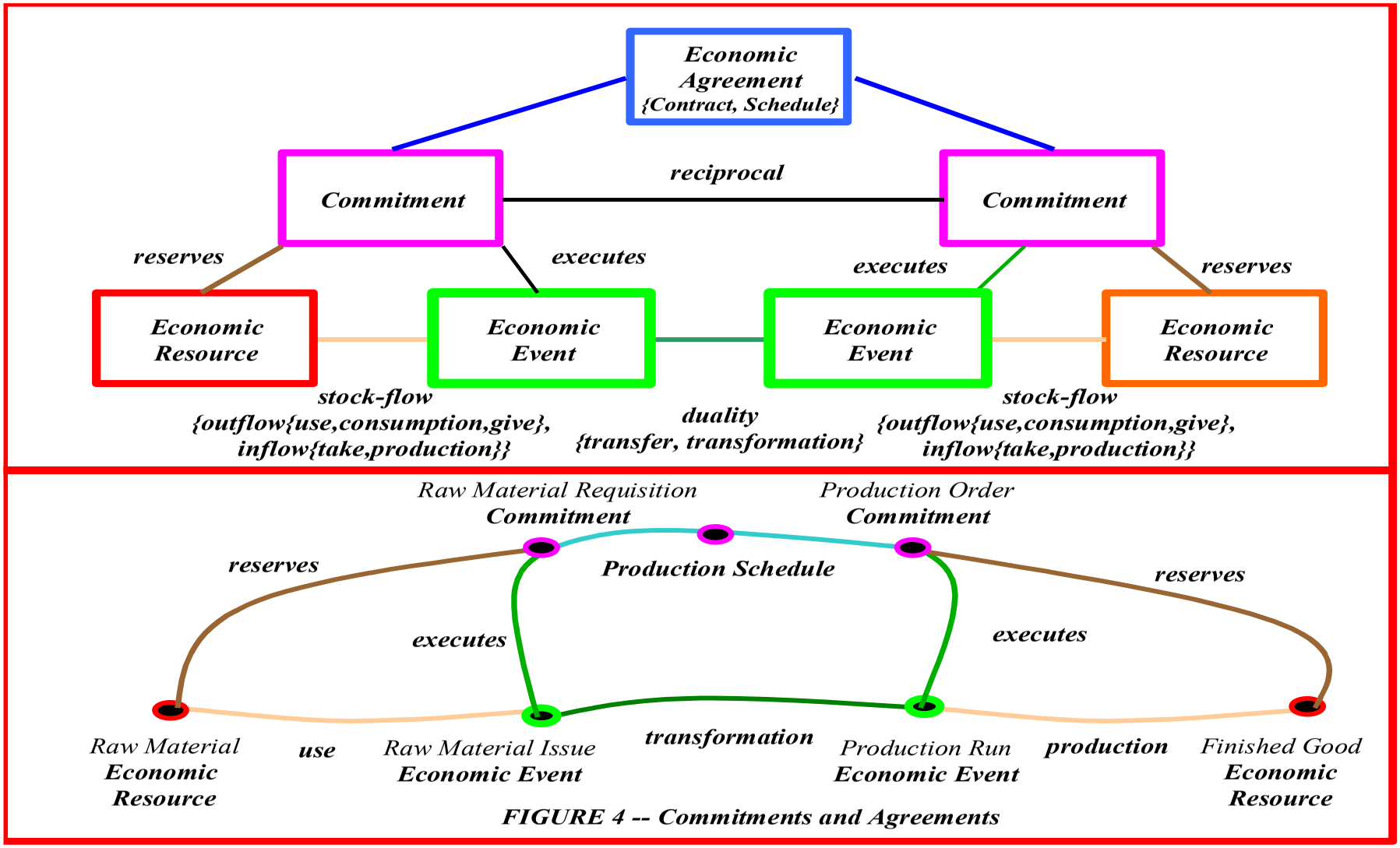


FIGURE 4 -- Commitments and Agreements

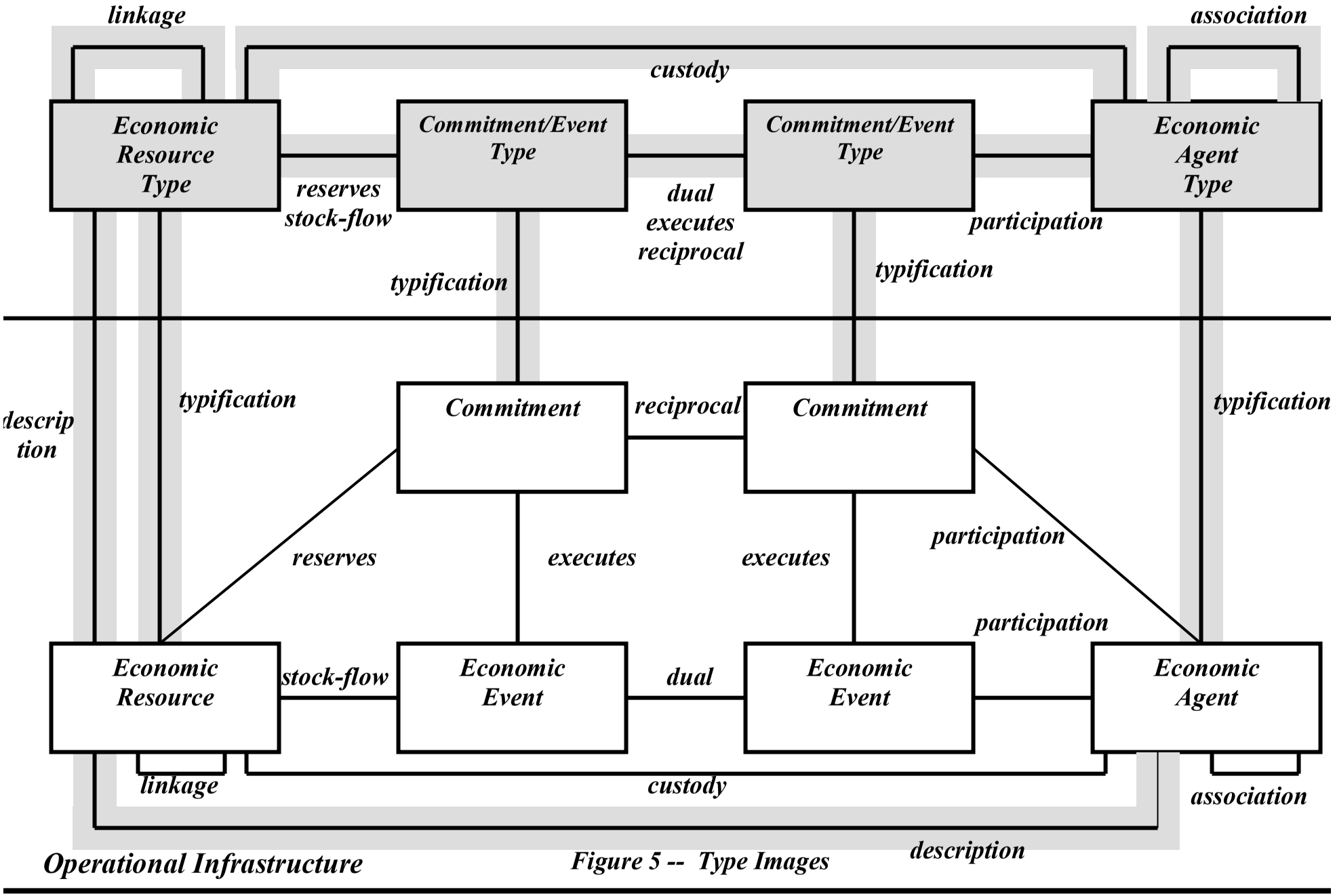


Figure 5 -- Type Images

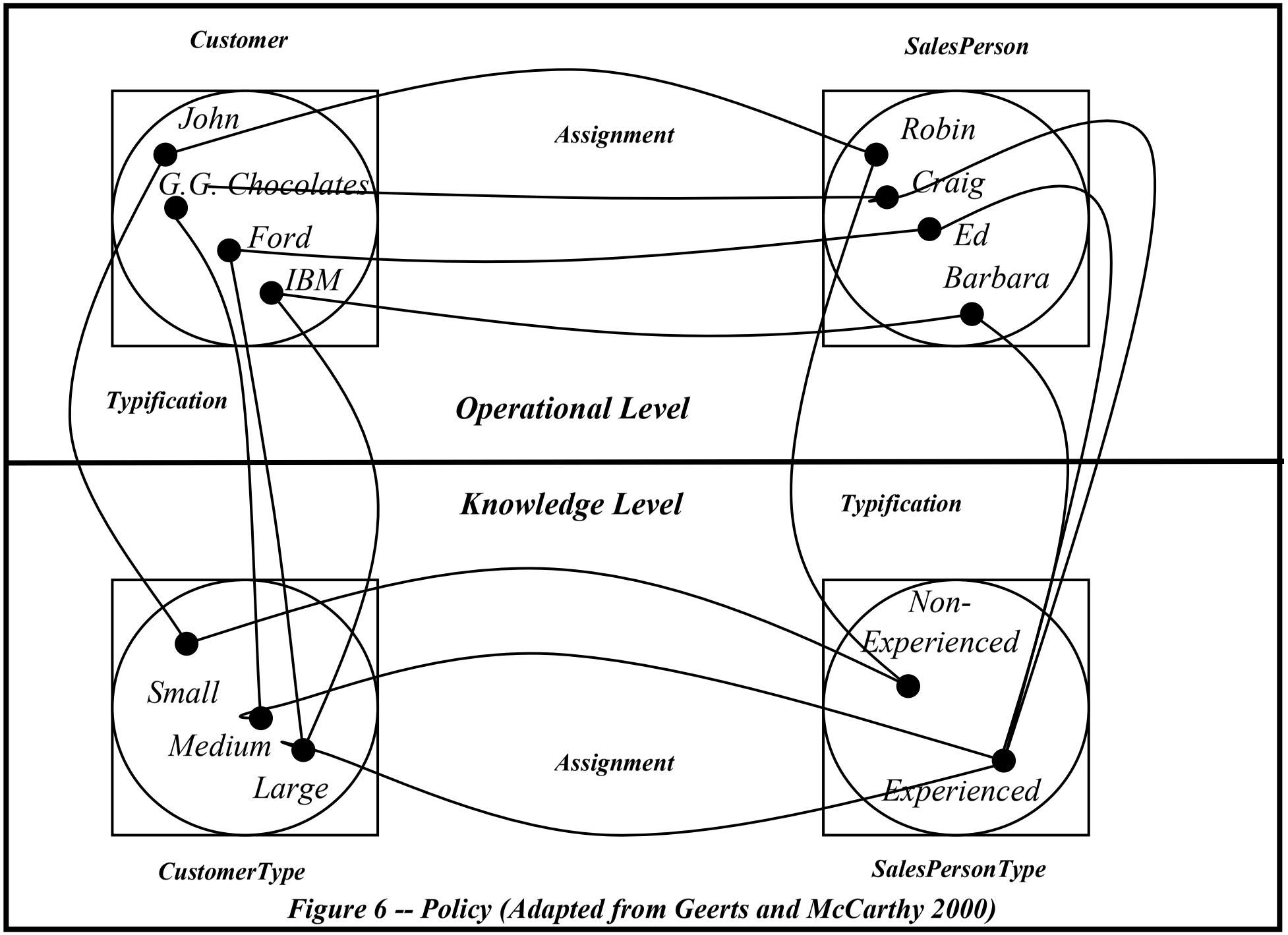


Figure 6 -- Policy (Adapted from Geerts and McCarthy 2000)

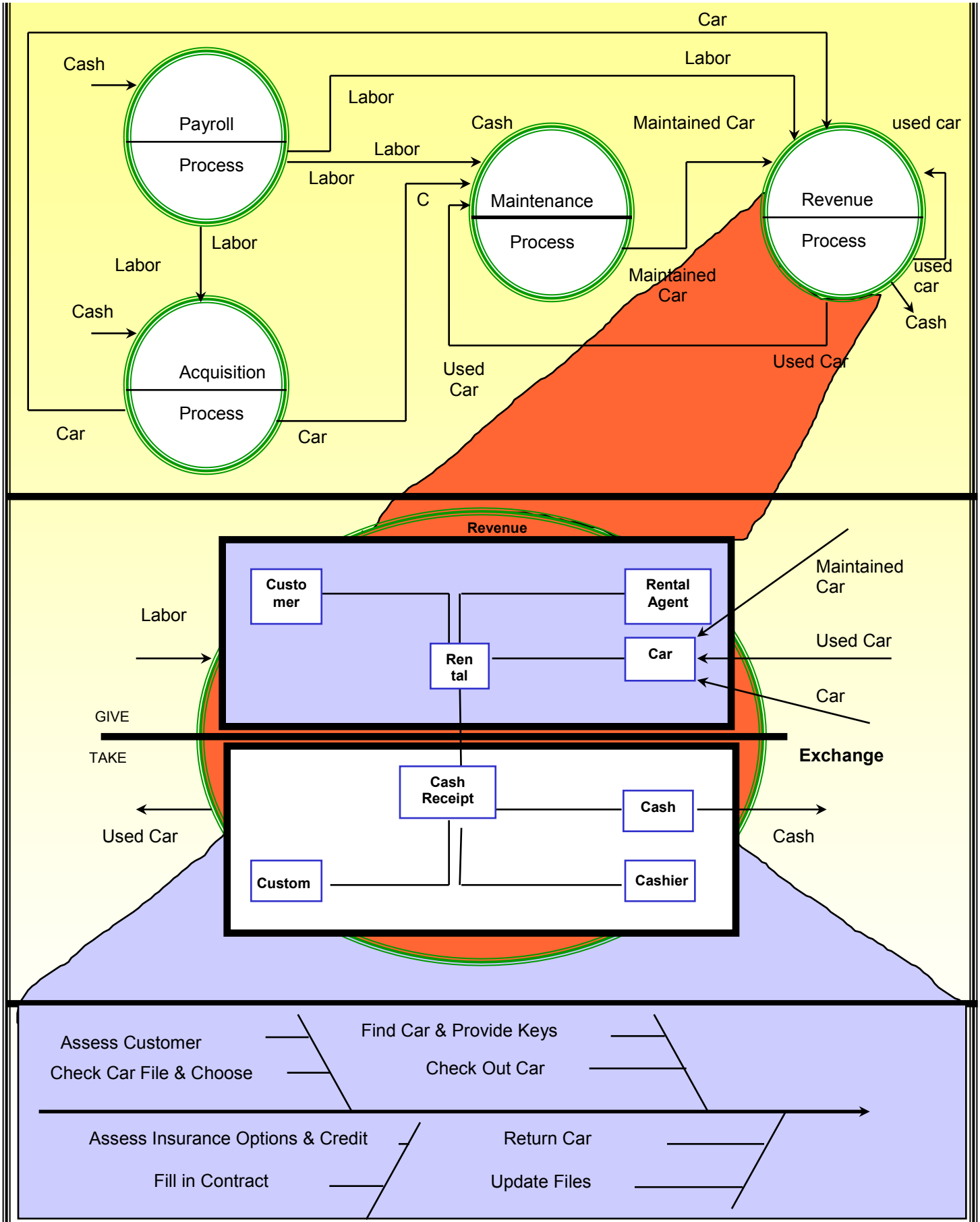


Figure 7 -- Three-Layer Architecture of REA Enterprise Information Systems

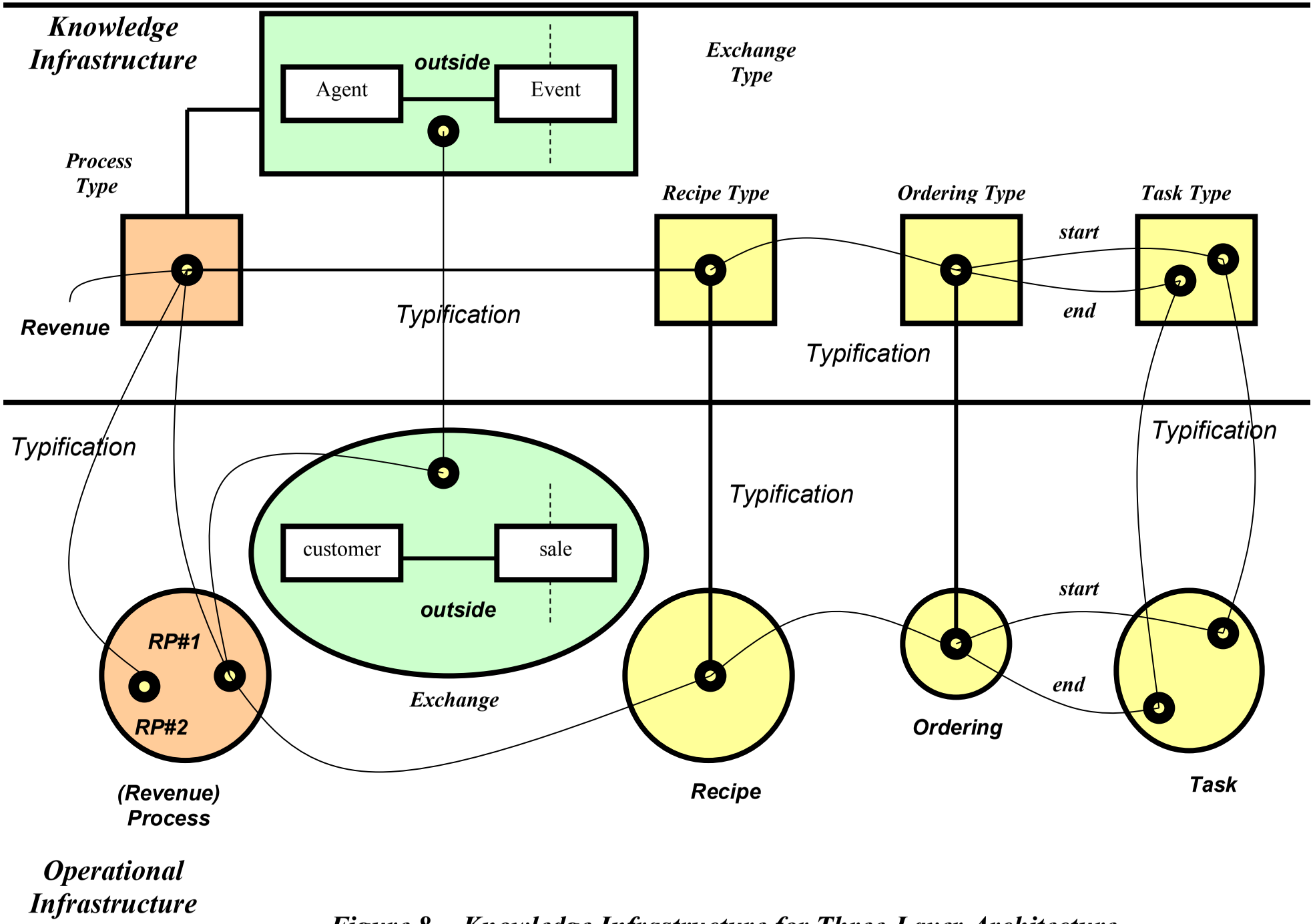


Figure 8 - Knowledge Infrastructure for Three-Layer Architecture