2002

VirtuE: Virtual Enterprises for Information Markets'

Alessandro D'Atri
LUISS University, datri@luiss.it

Amihai Motro
George Mason University, ami@gmu.edu

VIRTUE: VIRTUAL ENTERPRISES FOR INFORMATION MARKETS*

Alessandro D'Atri
CeRSI - LUISS "Guido Carli" University
datri@luiss.it

Amihai Motro
George Mason University
ami@gmu.edu

ABSTRACT

An essential part of a modern economy is an information market. In this market, information products are being traded in countless ways. Information is bought, modified, integrated, incorporated into other products and then sold again. Usually, the manufacturing of an information product requires the collaboration of several participants. A virtual enterprise is a community of business entities that collaborate on the manufacturing of new products. This collaboration is often ad hoc, for a specific product only, after which the virtual enterprise may dismantle. The virtual enterprise paradigm is particularly appealing for modeling collaborations for manufacturing information products, and in this paper we present a new model, called VirtuE, for modeling such activities.

1. INFORMATION MARKETS AND VIRTUAL ENTERPRISES

An essential part of a modern economy is an information market. In this market, information products are being traded in countless ways. Information is bought, modified, integrated, incorporated into other products and then sold again. Some information products are *elementary.* An elementary information product is created out of nothing; for example, a reading off an instrument, a photograph taken by a camera, or a new customer record added to a database. Often, however, information products are *derived* from other products. A weather report is assembled from multiple instrument readings, a news report requires the analysis of several sources, and even a photograph may be an enhancement of another photograph. It is possible that an information product is manufactured in its entirely by the same individual or business entity. More often, however, the production of an information product requires collaboration among several different participants.

A particular form of business cooperation that has attracted attention recently is that of a *virtual enterprise.* A virtual enterprise is a community of business entities that collaborate on the manufacturing of new products. The collaboration is often *ad hoc,* for a specific product only, after which the virtual enterprise may dismantle (indeed, former collaborators may become competitors). The members of a virtual enterprise often possess complementary skills and technologies whose combination is deemed necessary for the target product at hand.

The virtual enterprise paradigm appears to be suitable for collaborative productions of information products. In this paper we describe the VirtuE model for virtual enterprises whose core business is information products. Such a virtual enterprise may be set up to produce an *electronic publication,* with individual members providing services such as photo archives, news, layout, and proofing. A *market research* enterprise may be a collaboration among a credit bureau, a mailing list consolidator, an ar-

* This work was partially supported by the European Commission (FAIRWIS project IST-1999-12641) and by the Italian MIUR project on "Organization & Management of Information Systems in Virtual Enterprises within Digital Marketplaces".

chive of retail transactions, and a data-mining service. An *on-line library* may be a collaboration among many different information archives and a variety of information processors, providing specialized services such as document indexing, document retrieval and document ranking. In these examples, each of the members could, in turn, enlist the services of other members; for example, the mailing list consolidator could procure individual lists and then enlist the help of another service to integrate the lists while removing replications and resolving inconsistencies. The main features of VirtuE fall into four areas:

1. **Products:** VirtuE features two types of information products: *content,* which is an item of information, and *process,* which is an operation that modifies existing contents to produce new content. Enterprise members may offer both types of service. A simple global resource, called *dictionary,* is defined for coordinating knowledge about products among the participants.
2. **Manufacturing.** The set of products used or created by each enterprise member is stored in a local resource called *inventory.* For complex products (products that are created from more elementary products), *production plans* must be provided. A production plan specifies how contents and processes are combined to derive the new product.
3. **Transactions.** Since component products are often obtained from other enterprise members, a procurement mechanism is necessary. A catalog exchange process among the enterprise members establishes the *infrastructure* on which products are exchanged. Exchanges are executed in *two-phase transactions*: an inquiry followed by an order. Since production plans may branch into multiple alternatives, each requiring different transactions to import different components from different members, an *optimization* process selects the optimal plan.
4. **Customization.** VirtuE allows the definition of *performance indicators,* which are formulas that capture various quantitative characterizations of the virtual enterprise; for example, enterprise assets or interdependence level. Another feature of VirtuE are *constitutional rules*, which are constraints that express obligatory behavior; for example, require all participants to be of comparable magnitude, forbid internal competition, or promote a free market. Such rules may be used to create virtual organizations with different styles or flavors.

While we are not aware of any work on virtual enterprises for information markets, our work is at a junction of several active research areas, and we discuss here the most relevant five areas: (1) information marketplaces, (2) information brokering, (3) virtual enterprises, (4) workflow management in virtual enterprises, and (5) federated databases.

The importance of the information market, especially for electronic commerce, has been recognized for quite some time (Luffmann 1994) (Hagel and Armstrong 1997) (Grover and Teng 2001). It is within this important marketplace that VirtuE operates, advocating a specific type of collaborative organization to generate new products. VirtuE's goal of modeling information exchange and manufacturing is strongly related to information brokering. Predicting that federations of large number of information system will be cooperating to support information needs of users, (Kashyap and Sheth 1994) proposes an architecture consisting of information providers, information brokers and information consumers. The concept of virtual enterprise as a cooperative of independent entities that collaborate on the manufacturing of products has been around for decades. The interest of the information technology research community in this area dates only to the mid-1990s, with much of the work focusing on organizational issues, communication processes and information systems support (Mowshowitz 1997) (Monge and DeSanctis 1999) (Virtual Organization Net). Recently proposed paradigms to support virtual enterprises include the VEGA software platform (Suter 1998) and a methodology for fusing the separate business processes of enterprise members (Georgakopoulos et al. 1999) (D'Atri et al. 2001). VirtuE's concept of production plans may be considered a specialized form of workflow management (Georgakopoulos et al. 1995) (Worah and Sheth 1997) (Gal and Montesi 1999). Within this area, considerable attention had been given recently to the application of workflow management techniques to virtual organizations (Grefen and Hoffner 1999). Finally, in the area of data modeling, federated database models (Heimbigner and McLeod 1985) (Sheth and Larson 1990) (Seligman and Ker-

schberg 1993) (Prabhakar et al. 1993) may be considered predecessors of VirtuE. A federated database provides an environment in which information may be exchanged among autonomous or semi-autonomous participants. Federated databases, however, do not provide features for manufacturing new information products, and do not attempt to model the business aspects of information exchange.

A preliminary version of VirtuE is described in three sections. Section 2 describes the basic structures of the VirtuE model: participants, products, and an infrastructure that participants use to exchange products. This section also defines the global dictionary for coordinating knowledge about products. The manufacturing of new information products from existing products is discussed in Section 3, which introduces inventories and production plans. This section also discusses performance indicators and constitutional rules. Section 4 is devoted to operational issues. It defines two-phase transactions, and discusses the optimization of production plans. Section 5 concludes this paper with a brief summary and a list of subjects currently under investigation.

2. BASIC STRUCTURES

In this section we define the basic concepts of the VirtuE model. Each virtual enterprise consists of participants, products, and an infrastructure that the participants use for exchanging products. A global resource, called dictionary, is used to coordinate knowledge about products among the participants.

2.1 Participants: Members and Clients

We assume a community of independent members with shared interests. The members are independent in the sense that they remain autonomous and maintain their own assets. These assets include human, equipment or financial resources, as well as business expertise, such as knowledge about their production and delivery processes. Their shared interests are reflected in that they agree to cooperate with each other to produce joint products that are provided to common customers. After the community had been established it could evolve because existing members depart or new members join. Such evolution provides the enterprise with flexibility and allows it to adapt to new situations.

The set of members will be denoted M. An individual member will be denoted m_i, where i is a unique identifier of the member. A *client* is an entity outside the virtual enterprise that approaches the virtual enterprise to acquire a product.

2.2 Products: Contents and Processes

In practice, virtual enterprises may produce many different kinds of products. In VirtuE, we shall consider only *information* products, of the type that can be delivered over computer networks. Information products are provided by members of the enterprise to their clients. This provision is the ultimate purpose of an enterprise. Information products are also exchanged among the members of the enterprise in the production phase that precedes the provision of a product to a client. We distinguish between two basic kinds of information products: *content* and *process.*

Content is an information item. Examples include “a database of customers and the products they purchased in the year 2000,” “the codes of all stocks traded in the New York Stock Exchange and their closing values on March 31, 2001,” ”an image of the space shuttle landing in Kennedy Space Center on May 29, 2000,” “a stream of news items on the subject of sports,” and so on. A *request* for this kind of product describes the information needed; the *response* is information content that answers the description. To manage the potentially enormous number and variety of contents, we introduce *content types.* Each content is associated with one content type. Examples of content types are *Database, Image, Document,* and so on. Each content type C is associated with a sequence of *attributes,* $att(C)$. Each attribute A in $att(C)$ denotes a measurable aspect of all contents of this type, and has an associated *domain* of feasible values $dom(A)$. Let c be a content of type C, then $c[A]$ denotes the value of the attribute A for this content, $val(c)$ is the sequence of all attribute values of c (in correspondence with the sequence of attributes $att(C)$). The attributes $att(C)$ can be partitioned into several groups. One par-

ticular attribute, *Product_Code,* is associated with every content type and is used to identify products within the product offerings of a member. Next, a group of attributes is designated as *specificational.* These attributes are specific to individual content types and provide a description for each particular content of that content type. With these attributes, it should be possible to determine the essence of a product. For example, an *Image* type may have the attributes *Image_Format* and *Resolution* and a *Document* type may have the attributes *Document_Format* and *Author.* This group may also include attributes such as *Subject* or *Title.* The remaining attributes form the *optional* group. Examples include *Quantity, Size, Timestamp, Quality, Cost, Price* and *Member_Price.* A common optional attribute is *Description,* for describing products in notation such as natural language or keywords. Note that for *individual* content types, these attributes may be measured differently. We shall use $c[S]$ and $c[O]$ to denote, respectively, the specificational and optional attributes of a content c. Finally, an attribute is *mandatory* if each content of that type is required to have a valid value for that attribute. For attributes that are not mandatory *null* values are used wherever valid values are either unknown or inapplicable. *Product_Code* and the specificational attributes are mandatory.

The second basic kind of information product is *process*. A process is an operation that modifies given content to produce new content. Examples of processes include (1) aggregating a set of pictures in an *album,* (2) translating a document from one language to another, (3) analysis of financial data to produce stock market recommendations, (4) cleansing data (e.g., removing or correcting errors, resolving inconsistencies, and so on), (5) filtering data (i.e., separating the data into two parts: wanted and unwanted), (6) ranking a set of data items, (7) merging two lists while removing replications, and so on. A request for this type of product names the process and provides a set of input contents; the response is the output content. Usually, processing adds value to the original information. As with contents, we introduce *process types* to classify the different processes. For example, there could be several data compression processes, all belonging to a single process type *Compression.* Each process type P is associated with a sequence of input content types $C_1, \ldots, C_n$ and with an output content type C. When receiving contents $c_1, \ldots, c_n$, where c_i is of type C_i ($i = 1, \ldots, n$), a process p of type P produces content $p(c_1, \ldots, c_n)$, which is of type C. Process types also have their attributes. The attribute sequence of a process type P is denoted $att(P)$. Let p be a process of type P, then $p[A]$ denotes the value of the attribute A for this process. $val(p)$ denotes the sequence of all attribute values of p (in correspondence with $att(P)$). The *attributes* $att(P)$ are divided into the same groups: *Product_Code,* specificational and optional. Examples of specificational attributes are *Maximal_Error_Rate* or *Minimal_Output_Quality.*

In addition to the values in their associated domains, optional attributes may assume three special kinds of values. A *null* value is used when an attribute value is unavailable or inapplicable; a *multivalue* is a set of possible values; a functional value is a value that is a function of other attributes. For brevity, these special kinds of attributes are not discussed here in detail.

To summarize, the basic concepts of information products are: (1) content, content type and content attribute, and (2) process, process type and process attribute. A simple analogy that illustrates these concepts are computer files and file translators. Each file is associated with a specific file type (usually denoted by a suffix to its name), and has specific attributes (such as size, timestamps, and access permissions). Each file translator is associated with two file types: the source type and the target type. An example of a file translator is compression. An attribute for compression processes would be average compression rate, *Compression_Rate* may be a function of the *Density* of the input content.

2.3 The Global Dictionary

Of the six aforementioned concepts (content, content type, content attribute, process, process type and process attribute), all but content and process may be considered *intensional* information (content and process are *extensional* information). Intensional information is stored in an enterprise-wide resource called the *global dictionary.* This dictionary assures consistency of naming across the enterprise.

Formally, the global dictionary is a pair $(\boldsymbol{C}, \boldsymbol{P})$, where $\boldsymbol{C}$ is a set of content types and $\boldsymbol{P}$ is a set of process types. Each content type $C \in \boldsymbol{C}$ is described by a set of attributes $att(C)$ and each process type

$P \in \boldsymbol{P}$ is associated with a set of attributes $att(P)$. Each attribute $A \in att(C)$ or $A \in att(P)$ is associated with a domain $dom(A)$. Determining whether two products (content or process) are "the same" is not straightforward, and we define two different levels of identification:

1. **Identical products**: The attribute *Product_Code* uniquely identifies a product within offerings of individual members; hence, the combination *Member_Identifier:Product_Code* identifies a product across the virtual enterprise.
2. **Comparable products:** Two products (possibly from two different members) are *comparable* if they are of the same type and have the same specification; i.e., products e_1 and e_2 of type C are comparable if $e_1[S] = e_2[S]$. Intuitively, when two products are comparable, one could possibly substitute for another.

3. MANUFACTURING NEW PRODUCTS

The basic VirtuE paradigm is that members obtain information components from other members to manufacture new information products. This section explains how manufacturing is accomplished, using two new concepts: inventories and production plans. This section also explains how to create virtual enterprises possessing different characteristics by using constitutional rules and how to monitor their performance with performance indicators.

3.1 Inventories and Catalogs

Products (contents or processes) are exchanged by the members of the virtual enterprise. The method of exchange shall be explained later; at this point we note that the products used by each member can be classified according to two parameters.

1. **Native** or **Imported:** A native product is produced locally, whereas an imported product is outsourced from another member of the enterprise.
2. **Internal** or **Exported:** An exported product is provided to others, whereas an internal product is an interim product used only in the manufacturing of other products.

These parameters form four categories of products: native-internal, imported-internal, native-exported, and imported-exported.

Products (contents or processes) can also be classified as *basic* or *complex.* A content is complex if it is derived from other contents by some process; otherwise it is basic. A process is complex if it is a combination of other, more elementary, processes; otherwise it is basic. Note that imported products are always classified as basic; i.e., a complex product is always native. The set of products used by a member m are enumerated in an *inventory, Inv*(m). Each inventory entry e is described with six fields:

1. **Kind**: an indication whether the product is content or process.
2. **Type**: for a content, the *content type,* for a process, the *process type.* These types are taken from the global dictionary.
3. **Source**: an indication whether the product is native or imported.
4. **Target**: an indication whether the product is internal or exported.
5. **Composition**: an indication whether the product is basic or complex.
6. **Attribute values**: a sequence of values, corresponding to the attributes listed in the global dictionary for this content or process type.

The notation for the first five fields is similar to the notation for attribute values: $e[Kind]$, $e[Type]$, $e[Source]$, $e[Target]$, and $e[Composition]$.

A *service* is an offering of an information product by a member of the virtual enterprise for a price. Each member of the virtual enterprise advertises the services that it offers by means of a *catalog.* The catalog is simply the inventory items for which the target indication is "export." Note that a catalog may include services that require the offering member to obtain assistance from other members of the enterprise. This is referred to as *subcontracting* (outsourcing). Consequently, a change in a catalog (such as a price increase) may propagate to other catalogs.

Each member *distributes* its catalog to a subset of the members of the enterprise, and receives catalogs from other members. A member must either send its catalog to or receive a catalog from at least one other member (otherwise this member would not be able to participate in any activity of the virtual enterprise). This distribution creates the *infrastructure* of the virtual enterprise, as it describes the various channels of procurement. Any change to a catalog, such as a price increase, requires the redistribution of the catalog. The set of members to which a member $m_i \in M$ distributes its catalog is a subset D_i of M. The infrastructure is a subset D of $M \times M$.

3.2 Production Plans

For each complex content or process in the member's inventory there must be a *production plan* (a manufacturing formula). These production plans are expressed in terms of other contents and processes. In addition to describing the structure of complex products, production plans also assign their output the appropriate attribute values.

In describing production plans, we use the following notation; c and p identify a content and a process, respectively. If the product is native, then c and p are product codes from this member's inventory. If the product is imported, then c and p are *external* product codes; i.e., each is a combination of a member identifier and a product code from that member's catalog.

Production plans must be provided for each *native complex* content. Native basic contents and imported contents are simply referenced by their product codes. The production plan for native complex content c is a combination of a process p and attribute assignments as follows.

$$c \leftarrow p(c_1,\ldots,c_n)$$
$$c[A] \leftarrow \phi_A(val(p),val(c_1),\ldots,val(c_n)) \quad \text{for every attribute } A \text{ in } att(C)$$

The meaning of the first line is that applying the process p to a sequence of contents $c_1, \ldots, c_n$ produces the content c. The second line describes a set of assignments ϕ_A, one for each attribute A of c. Each ϕ_A assigns a value for the attribute A based on the attributes of the input contents $c_1, \ldots, c_n$ and the process p. Note that each input content c_i may be either native or imported. If an input content c_i is native and complex, then another production plan must be provided for c_i. Hence, the production plan for c may recursively involve other production plans (it must be, of course, free of cycles). Note also that a content c may have *several* production plans, allowing for alternative manufacturing processes.

There is a special production plan called *substitution* that involves no processing: $c \leftarrow c'$. Substitution allows one content to substitute for another. For example, using different substitution formulas with the same left-hand-side, one may specify alternative imports for the same content. Finally, note also that the process p may be native or imported. When p is imported from member m, then after $c_1, \ldots, c_n$ are materialized, they are sent to m, who subsequently sends back the content c. Note that attribute values are included with the input and output contents.

Similarly, production plans are defined for processes. Using such production plans, it is possible to establish the *cost* of every content or process. It is the *price* paid for externally procured contents and processes, plus the *cost* of internally procured products and processes. The price of this would usually be higher (the difference is *profit*). The new cost would be determined by the assignment ϕ_{cost}. Consider a process $c \leftarrow p(c_1, \ldots, c_n)$ and assume that p is a native process. In this case, ϕ_{cost} would be:

$$c[Cost] = p(Cost) + \sum_{\substack{c_i \\ c_i[Source]='native'}} c_i[Cost] + \sum_{\substack{c_i \\ c_i[Source]='import'}} c_i[Price]$$

3.3 Performance Indicators and Constitutional Rules

Using the notation developed earlier we may express various indicators and rules. Indicators are quantitative characterizations of the enterprise, whereas rules express obligatory behavior.

Performance indicators may be classified in a three-level hierarchy: (1) Product-specific, (2) Member-specific, and (3) Enterprise-wide. Product-specific indicators characterize individual products. Member-specific indicators characterize the performance of a member; often, they are defined by summarizing product-specific indicators. Similarly, enterprise-wide indicators characterize the performance of the entire enterprise; often, they are defined by summarizing member-specific indicators. Product-specific indicators could be considered *derived attributes.* A simple example is *profit*, the difference between the price and the cost of a product:

$$Profit(e) = e[Price] - e[Cost]$$

Another example is *exclusivity:* the number of comparable products available throughout the enterprise. Let $e \in Inv(m)$, $e[Target] = 'export'$, then

$$Exclusivity(e) = \left|\{e' \mid e' \in Inv(m') \wedge m' \neq m \wedge e'[Target] = 'export' \wedge e'[S] = e[S]\right|$$

A product e is *exclusive* if $Exclusivity(e) = 0$. A third example is *external-dependence* (import-dependence), which measures the ratio of the cost of imports used in a product to the price of the product. Let $Imp(e, q)$ be the set of imported products used in a production plan q for product e. Then

$$Dependency(e) = \min_q \frac{\sum_{e' \in Imp(e,q)} e'[Cost]}{e[Price]}$$

Additional indicators may be defined similarly: Product *robustness* is the number of alternative production plans, product *complexity* is the average depth of its production plans or the average number of components used, and so on. An example of a member-specific indicator is *breadth*, which measures the number of export products in the member's inventory (i.e., the size of its catalog):

$$Breadth(m) = \left|\{e \mid e \in Inv(m) \wedge Target[e] = 'export'\}\right|$$

Using product exclusivity, one could measure *member exclusivity* as the average exclusivity indicator of this member's products. The *assets* of a member (the *total value* it adds) can be defined as the total price of all the items in its inventory marked "export" less the total cost of the items marked "import":

$$Import(m) = \sum_{\substack{e \in Inv(m) \\ e[Source]='import'}} e[Cost]$$

$$Export(m) = \sum_{\substack{e \in Inv(m) \\ e[Target]='export'}} e[Price]$$

$$Assets(m) = Export(m) - Import(m)$$

An important indicator is the *level of interdependence* (cooperation) among the members of a virtual enterprise. This may be measured as the ratio of imports to assets: the higher the ratio the more dependent is the member on other members:

$$Depend(m) = \frac{Import(m)}{Assets(m)}$$

Enterprise-wide interdependence may be expressed by averaging the individual indicators:

$$Depend = \frac{1}{n}\sum_{m \in M} Depend(m)$$

If *Depend* falls below a threshold, it may be advisable to reorganize the enterprise or dismantle it.

The global behavior of a virtual enterprise is governed by a set of enterprise *rules*, which express the *constitution* of enterprises and give them individual characteristics. Some illustrative examples follow.

We already mentioned that each member must be involved in at least one catalog exchange:

$$\forall m \in M \quad \left|\{m' \mid m' \in M \wedge ((m,m') \in D \vee (m',m) \in D)\}\right| \geq 1$$

There may be a rule that requires members to give fellow members preferred treatment over clients; that is, a member cannot offer products to clients at prices lower than those it offers to members:

$$\forall m \in M \quad \forall e \in Inv(m) \quad e[Price] \geq e[Member\_Price]$$

A rule may be defined to disallow "dumping," the practice of selling items below their cost; that is, the price charged for must exceed the cost incurred in producing the item (an example of cost calculation was given earlier). As a final example, consider a rule that establishes *product exclusivity*; i.e., there are no comparable products in the virtual enterprise.

$$\forall m_1, m_2 \in M \quad \forall e_1 \in Inv(m_1) \quad \forall e_2 \in Inv(m_2)$$
$$e_1[S] = e_2[S] \Rightarrow m_1 = m_2 \wedge e_1[Product\_Code] = e_2[Product\_Code]$$

Recall that identical products are comparable; this rule implies that comparable products are identical.

4. TRANSACTIONS

The infrastructure defined by the distribution of catalogs supports the operations of a virtual enterprise. The basic unit of operation in a virtual enterprise is a *transaction.* A transaction begins when a request for an advertised service (content or process) is sent from one participant to another, and terminates when the request is satisfied. There are two types of transactions in a virtual enterprise.

- **External transaction**. An external transaction is a request for a service that is submitted from a client to one of the members of the virtual enterprise. The member processes the request and provides a solution. A member of the virtual enterprise who processes an external transaction acts in a role of a *service provider*.
- **Internal transaction**. To satisfy an external transaction, a service provider may decide to purchase a service from another member. Such transactions are called *internal transactions* or *subcontracts.* A member of the virtual enterprise who processes an internal transaction acts in a role of a *subcontractor.* Sub-contracting is related to information brokering.

The execution of external transactions is the ultimate purpose of the virtual enterprise. Each member of a virtual enterprise may act as a service provider on some transactions and as a subcontractor on other transactions. To initiate a transaction, the requester must provide the exact specifications of the service required. Because products may have attributes that are multivalues or functions, a preliminary exchange of information may be necessary. Such exchanges are called *inquiries.* Actual requests for service are called *orders.* Hence, in general, products are traded with *two-phase* transactions.

An inquiry is a request for additional information, and is necessary when some of the attributes of the product are multivalued and other attributes are functionally dependent on them. In an inquiry, the requester specifies values for some attributes. In response, the provider specifies values for all other attributes. Note that to determine its responses, the provider may need to initiate inquiries to its subcontractors. After making the necessary inquiries, a participant may issue an order. Each order must include (1) the code of the product, (2) specific values for its multivalued attributes, if any, and (3) the

input contents (and their attributes), if the product is a process. Once a member receives an order (either external or internal), it must put together a plan to deliver the product as specified.

If every product (content or process) had only one set of specifications (i.e., without multivalued attributes), and if every product had a unique production plan, then the fulfillment of orders would be a straightforward process. By allowing a product to have different specifications and alternative production plans (including multiple options for importing component products), the fulfillment of orders becomes a process that requires *optimization*. Note that we assume that when a service provider advertises a product with multivalued attributes, there must be appropriate production plans to manufacture products that will meet *any* of the advertised attributes. Some issues involved in such optimization are discussed below.

The resources necessary for putting together an execution plan may be available from multiple sources, thus suggesting various alternatives that must be considered. For example, content may be available from multiple sources with different attributes; e.g., a member m who needs content c in quantity q may have the options of (1) buying c with in quantity q, or (2) buying c in higher than needed quantity q' but at a lower price. As another example, content may be derived in different ways; e.g., a member m who needs content c may have the options of (1) buying c from member m_1, or (2) buying content c' from member m_2 and using the services of member m_3 to transform c' to c.

In general, each service provider adopts an attribute (or a weighted combination of attributes) that would serve as its *optimization target*. Given an inquiry or an order, the service provider would derive all the possible production plans, calculate the value of the optimization target for each plan, and choose the optimal plan. While optimization is defined through an exhaustive process that considers every possible plan, in practice, when this approach is infeasible due to a large number of plans, heuristic optimization methods should be developed. Typically, a service provider would attempt to optimize the *cost* of production, while satisfying all the product specifications. That is, among the production plans that answer the specifications, the provider would choose the plan with minimal cost.

5. CONCLUSION

We presented a preliminary version of VirtuE, a model for virtual enterprises that collaborate on the manufacturing of information products. Some of the features of VirtuE that make it suitable for this purpose are: (1) Two types of information products, *content* and *process,* and a *global dictionary* for knowledge coordination. (2) *Inventories* and *production plans* for expressing the manufacturing processes of information products. (3) *Catalog exchanges* and *two-phase transactions* for enabling the exchange of information products. (4) *Constitutional rules* and *performance indicators* for creating enterprises with different characteristics and for monitoring their behavior. There are many possible research directions to follow up the preliminary results presented in this paper, and we mention here briefly four such directions.

(1) **Enduring products and transactions**. The model we described considered "one-time" products and transactions; that is, each transaction traded a single, already available commodity. Many information products are manufactured *continuously* from *streams of information.* Accordingly, transactions are contracts for the continuous supply of services. The proper extension of VirtuE to model continuous services is currently under investigation. (2) **Tracking performance over time.** The current model does not track the actual performance of a virtual enterprise over time. Such tracking would permit new performance indicators, such as *demand* or *profit.* (3) **Dynamic reorganization.** An important advantage of virtual enterprises is their ability to adapt quickly to changing markets. By monitoring the actual performance of an enterprise, including changes in the types or quantities of products ordered, it would be possible to prescribe changes in membership, production plans and infrastructure that would improve the overall performance of the enterprise. (4) **Extended cost modeling.** Through the use of performance indicators and constitutional rules, VirtuE attempts to model the business aspects of information exchange and manufacturing. A possible extension is to consider cost models such as (Appel and Behr 1998) (Belgradek 1998).

Once the VirtuE model has been finalized, the next steps would be to develop a functional specification for a software environment based on this model that will support the activities of a virtual enterprise for information markets, and subsequently to develop a prototype system in accordance with these specifications.

REFERENCES

APPEL, W. and R. BEHR (1998). Towards the Theory of Virtual Organisations: A Description of their Formation and Figure. *Virtual-Organization.Net Newsletter,* 2(2), 15-36.

BELGRADEK, B. et al. (1998). Maximizing Seller's Profits for Electronic Commerce. In *Proc. IFIP/GI Working Conference on Trends in Distributed Systems for Electronic Commerce*, Lecture Notes in Computer Science No. 1402, Springer-Verlag, Berlin, 26-38.

D'ATRI, A., A. SOLVEBERG, L. WILLCOCKS (Eds.) (2001). *Open Enterprise Solutions: Systems, Experiences, and Organizations*. Luiss Edizioni, Rome.

ERBEN, K. and K. GERSTEN (1997). Cooperation Networks Towards Virtual Enterprises. *Virtual-Organization. Net Newsletter,* l(5), 12-22.

GAL, A. and D. MONTESI (1999). Inter-Enterprise Workflow Management Systems. In *Proc. 10th International Workshop on Database and Expert Systems Applications*, 623-627.

GEORGAKOPOULOS, D. et al. (1995). An Overview of Workflow Management: from Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases,* 3(2), 119-153.

GEORGAKOPOULOS, D. et al. (1999). Managing Process and Service Fusion in Virtual Enterprises. *Information Systems: Special Issue on Information Systems Support for Electronic Commerce,* 24(6), 429-456.

GODART, C. et al. (1999). Coo: a Workflow Operator to Improve Cooperation Modeling in Virtual Processes. In *Proc. 9th International Workshop on Research Issues on Data Engineering,* 126-131.

GREFEN, P. and Y. HOFFNER (1999). Crossflow: Cross-Organizational Workflow Support for Virtual Organizations. In *Proc. 9th International Workshop on Research Issues on Data Engineering, 90-91*.

GROVER, V. and T. C. TENG (2001). E-commerce and the Information Market. *CACM,* 44(4), 79-86.

HAGEL, J. and A. ARMSTRONG (1997). *Net Gain. Expanding Market Through Virtual Communities.* Harvard Business School Press, Boston.

HEIMBIGNER, D. and D. MCLEOD (1985). A Federated Architecture for Information Management. *ACM Transactions on Office Information Systems,* 3(3), 253-278.

KASHYAP, V. and A. P. SHETH (1994). Semantics-Based Information Brokering. In *Proc. 3rd International Conference on Information and Knowledge Management,* 363-370.

LAUFMANN, S. (1994). The Information Marketplace: The Challenge of Information Commerce. In *Proc. 2nd Conference on Cooperative Information Systems*, 147-157.

MONGE, P. and G. DESANCTIS (Eds.) (1999). Special Issue on Virtual Organizations. *Organization Science,* 10(6).

MOWSHOWITZ, A. (Ed.) (1997). Special Section on Virtual Organizations. *Communications of the ACM,* 40(9), 30-64.

PRABHAKAR, S. et al. (1993). Federated Autonomous Databases: Project Overview. In *Proc 3rd Research Issues on Data Engineering: Interoperability in Multidatabase Systems*, 216-219.

SELIGMAN, L. J. and L. KERSCHBERG (1993). Knowledge-Base/Database Consistency in a Federated Multidatabase Environment. In *Proc 3rd Research Issues on Data Engineering: Interoperability in Multidatabase Systems*, 18-25.

SHETH, A. P. and J. A. LARSON (1990). Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases. *Computing Surveys,* 22(3), 183-236.

SUTER, B. (1998). A Cooperation Platform for Virtual Enterprises. In *Proc. VoNet Workshop on Organizational Virtualness, 155-164.*

VIRTUAL ORGANIZATION NET. Electronic Journal of Organizational Virtualness (ISSN 1422-9331). http://www.virtual-organization.net.

WORAH, D. and A. P. SHETH (1997). Transactions in Transactional Workflows. In *Advanced Transaction Models and Architectures*, Kluwer Academic Publishers, 3-34.