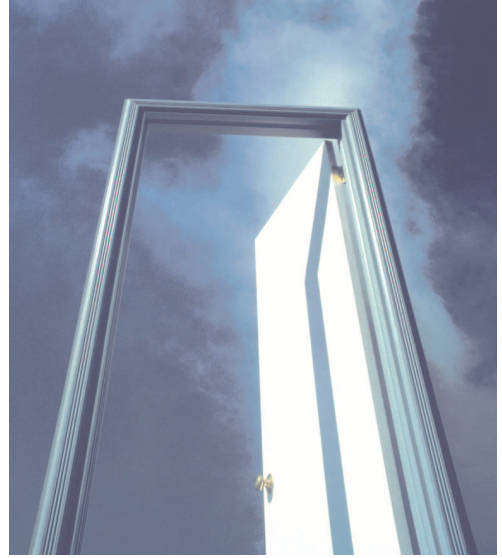


ERP5 combines an object database and innovative open-source ERP components targeted at small and medium-size companies.

Jean-Paul Smets-Solanes
Rogério Atem de Carvalho



ERP5: A Next-Generation, Open-Source ERP Architecture

When someone says enterprise resource planning (ERP), most IT professionals think of the expensive, complex, and difficult-to-implement commercial products that were the rage a few years ago. Although many large corporations did reap tremendous cost savings from the implementation of such systems, an average implementation cost counted in the millions of dollars; this has prevented ERP systems from spreading to small and medium-sized businesses. After ERP deployment, its “blackbox” nature prevents from understanding and eventually improving the business processes it implements, leaving some important business decisions to the software publisher rather than to the corporate manager, preventing scientific researchers from getting involved in the management innovation.

This situation provides much of the motivation for our architecture, ERP5, which offers several advantages for business. All ERP5 tools are open source, so are free and have openly available source code that a business can change to suit its processes. ERP5 incorporates, from scratch, advanced concepts such as object-oriented data-

bases, a content management system, synchronization, variations, workflows, and a method to model and implement business processes. ERP5 is also a Web site where researchers can

share innovation on management techniques and their implementation through software.

In 2001, two companies initiated the ERP5 project: Nexedi, a Zope service provider in France (Zope is a well-known open-source application server), and Coramy, a European apparel manufacturer. They aimed to develop a set of ERP software components for small and medium-sized companies. In addition to source code, the project also produced education material and a clearly defined theoretical model. To fit the needs of smaller companies, they also designed ERP5 for distribution across distant sites with slow and unreliable Internet connections.

ERP5 MAIN COMPONENTS

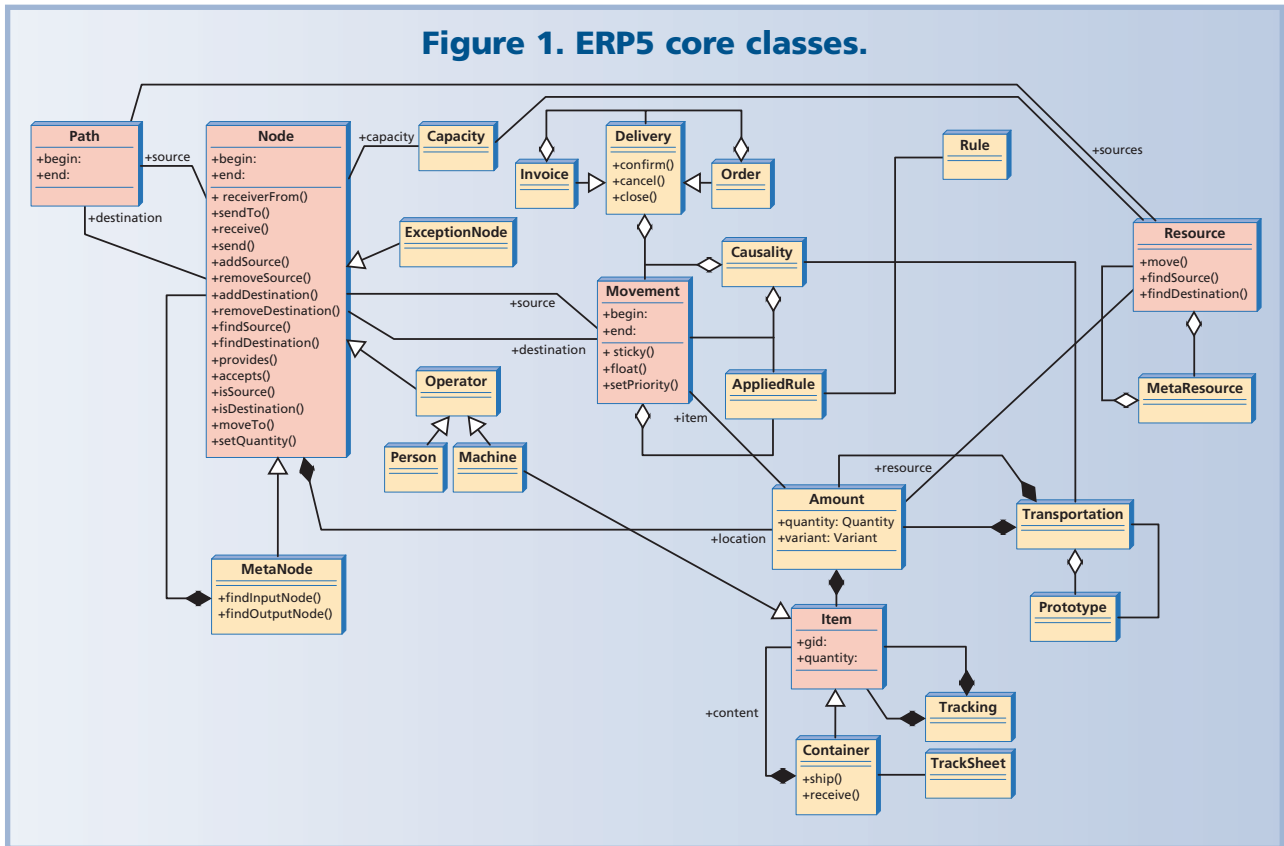
Like other ERP systems, ERP5 uses components as the basis for the system. All functions derive from or depend on only five basic concepts.

- *Planning.* ERP5 offers common ERP functions like production planning and control, materials management, finances, accounting, billing, budgeting, human resources, and the compatible transactions.
- *Customer Relationship Management.* ERP5 tracks customer relations (like meetings, sales prospects, and phone calls) and provides a reporting engine to classify customers into categories (by country, size, and market segment, for example).

Inside

Resources

Figure 1. ERP5 core classes.



- **Content Management System.** ERP5 can act as a content management system for an intranet. The system can store, index, and classify all documents. The ERP5 content management system can retrieve a document based on the combination of a category (production incidents, for example) and a keyword (such as “overheat”), which helps to capitalize on knowledge within the company. This way, ERP5 can deal with both structured and non-structured information.
- **E-business.** ERP5 implements B2C and B2B systems. Thanks to workflow modeling, a sales process can manage all the interactions between the online shop and customer, including e-mail exchanges. Using this component, it is possible to automate typical transactions, such as when a customer returns a purchase.
- **Groupware.** ERP5 integrates with groupware products, including those for e-mail, agendas, directories, shared workspaces, IP telephony, IP conferencing, and instant messaging.

ERP5-SPECIFIC FEATURES

ERP5 is multilingual and deals with many types of currency; it works across several companies and users. It also incorporates the unified model of business flows in Figure 1, which maps all corporate information into five basic concepts (thus the name ERP5). It also uses the concepts

of variation, meta-planning, and information synchronization.

To support these features, on the technological side, ERP5 offers a new mapping technology for object or relational DBMS (database management system) integration, a new active-messaging technology for interaction modeling, and a new synchronization technology for electronic data interchange.

Variation

ERP5 can represent possible variations for a given resource, such as color, size, and speed. Variations are very useful in industries such as computers, to specify memory size, disk size, or processor speed; apparel, to specify color and size; and cars, to specify color, engine, or options. Variations define complex resources that result from successive transformations. ERP5 represents each transformation as a collection of transformations, some of which apply to all variations. Others apply only to certain variations.

Variations use only a single resource descriptor and a collection of options to define many configurations of a given product. This strategy avoids the creation of a complex taxonomy, a huge number of database records, and the tracking of a product number for each variation of the same product. Although used extensively in the apparel

Table 1. Zope features.

Feature	Description	Advantages for ERP5 project
Object database	Zope relies on the Zope Object Database (ZODB) to store objects, and for Zope, everything is an object. ZODB uses a folder metaphor to organize objects and components. Fully transactional, ZODB works with the notions of version and history, which permits the change of some user data on a production system without interfering with other users. These concepts also permit the viewing of all transactions on an object that have occurred since its creation.	Advanced information systems, especially decision support systems, must deal with structured (database based) and unstructured (documents and images) information. ZODB facilitates the storage, search, and categorization of both of types of information through objects.
Object publishing	Objects and methods are accessible through a URL (uniform resource locator). Remotely calling a method on an object is possible by sending an HTTP (Hypertext Transport Protocol) request with method parameters provided through a get or post method. Using Extensible Markup Language and Remote Procedural Call can produce the same results.	A totally Internet-based system like ERP5 must have everything publishable through the Web.
Access control lists	Zope includes a security model that associates each method of each component class to a security group. The security model assigns each user one or more roles. Each object in a ZODB has a security access control list that defines which roles can access which security group's methods.	Provides the fine-grained security control that Internet-based systems require.
Products	It's possible to extend Zope with software components called products in the Zope terminology. Developing a new product is easy, and the Zope community has already developed and maintains hundreds of products.	ERP systems are big and have long life cycles; thus, facilities to extending their behavior are a must-have for any ERP development environment.
Workflow engine	Zope implements entity-based workflows. An entity—a business object in ERP5—always has a state associated with it. It also has a set of possible transitions that could occur if certain conditions are met. When an entity is in a given state the authorized actors have a set of available actions they can perform on the entity.	Workflows are vital in modeling business processes. Unifying entity-based workflows with objects makes ERP5 provide a clear structure for rapid application modeling and development.

industry, the concept of variations is not well-known or used in other industries.

Most commercial ERP software does not support variations, but we believe variations should be at the heart of any modern ERP system. Variations will be essential to creating product lines according to a customer's needs without creating thousands of product identifiers. This mass-customization concept manages customer needs without losing the benefits of standardization.

Meta planning

We base ERP5 on a model that can associate anything to a category. Examples include a category of resources (such as service, raw material, skill, or money) or a category of organizations (such as a group of companies, a group of people, or a retail chain). ERP5 can manipulate these categories just as it manipulates resources and organizations. This means that users can plan resources at any given level, whether for a group of companies or for a retail chain. Resource planning can be detailed (the company expects to sell 10,000 items of model number 223311 to customer

C) or general (the company expects to sell 10,000 of product X to small retailers). In this example, product X is a meta-resource and small retailers are a meta-organization—a metanode in ERP5 terminology. ERP5's meta feature helps manage a group of companies that belong to a common holding; it can also manage *factoring partnerships*—partnerships where one company asks a few other companies to cooperate and together build a product.

Information synchronization

We designed ERP5 to function on multiple sites with low-quality Internet connectivity, so each site must be able to run by itself in the case of a network failure. ERP5 implements distribution by using synchronization and an extension of the SyncML protocol. For example, suppose a stock A and a stock B located on two production sites AA and BB exchange some resource through a business rule. If this rule requires site BB to know the stock values of A, ERP5 will create a local representation of stock A on site BB and synchronize data between this local representation and the real values of stock A. Site BB will then

process its business rules on the local representation of stock A rather than on the real A itself. We believe synchronization is the next generation of EDI (electronic data interchange). It defines a subset of data that two companies want to share. We call this synchronization a common business vision.

Synchronization implements all standard EDI features. Order synchronization is equivalent to EDI order transmission. Model synchronization is equivalent to EDI model transmission. Although theoretically equivalent to EDI, synchronization is much easier to implement than an EDI approach. PDA users are perfectly aware of this: Every day they synchronize their personal information (contact lists, agendas, tasks, and expenses) with a corporate server. Such a transfer of information is rare in EDI, although in theory not difficult to implement.

USING AND EXTENDING THE ZOPE INFRASTRUCTURE

ERP5 uses Zope as its foundations. Zope is an open source Python-based application server and content management system. Zope runs on major operating systems such as Linux, FreeBSD, many Unix flavors, MacOS X, and Windows.

On the hardware side, Zope runs on many platforms, including x86 PCs, Macintoshes, Sun workstations, and IBM mainframes. It also provides Zope enterprise objects (ZEO), which are designed for applications that answer thousands of requests a day. ZEO allows an application to run on more than one computer, providing clustering and load-balancing capabilities. Running Zope on multiple computers allows even distribution of requests; administrators can add more computers as the number of requests grows. Further, if one computer fails or crashes, other computers can still service requests. Table 1 shows Zope features that justify its use in environments that service a high volume of requests and must remain in constant operation.

ERP5 uses, extends, or provides five Zope products:

- page templates for Web user interface presentation logic,
- workflows for rapid development of decision processes,
- a formulator for rapid form development,

Figure 2. Zope organizes objects into folders.

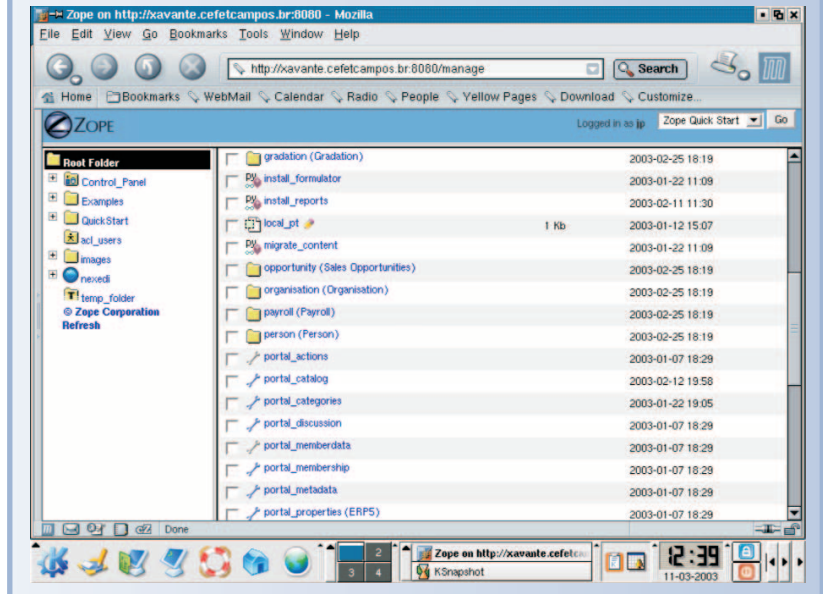
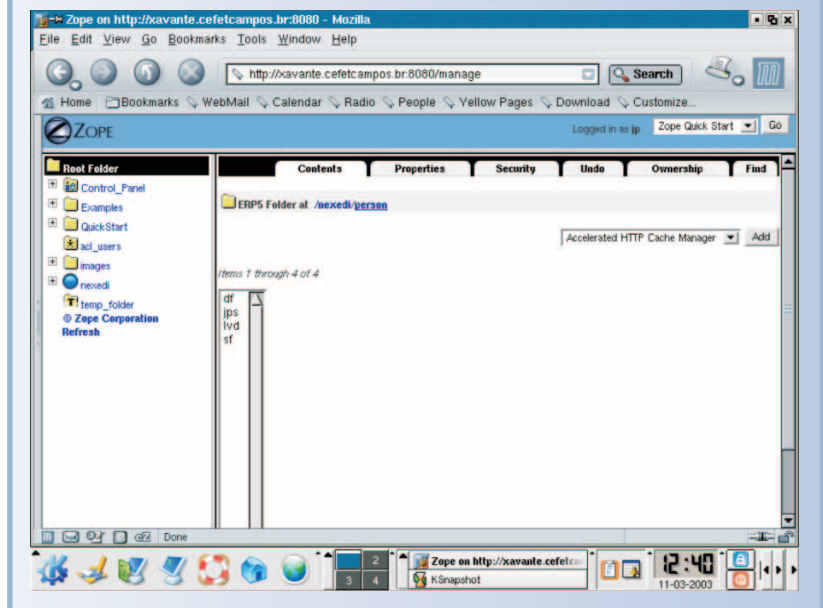


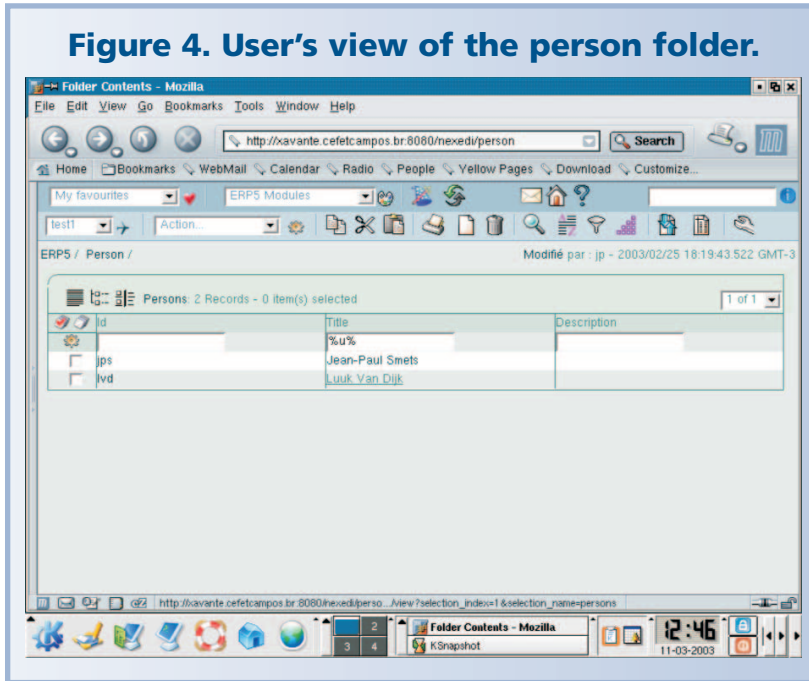
Figure 3. System administrator's view of the person folder through the ZMI.



- the ZSQLCatalog component to provide the capability for SQL (Structured Query Language) queries to the ZODB. ZSQLCatalog is an object-relational indexing component that allows searching the ZODB using SQL queries. It is crucial in terms of features, because it allows implementing complex queries in Zope, and performance, because it allows you to manage, for example, 10 workflows with 20 states and 10 roles each. This would be impossible with the standard Zope indexing system, and
- the XMLObjects component to add SyncML synchronization and autonomous activities to Zope objects.

Figures 2 and 3 show how system administrators organize

Figure 4. User's view of the person folder.



and view the objects through the ZMI (Zope management interface). Figure 2 shows how Zope organizes objects into folders. This ZMI snapshot shows a partial view of an ERP5 installation's main folder, called *nexedi*, as the right-hand panel indicates. Among other objects, you can see subfolders like *organization* and *person*. Figure 3 shows a view of the person folder through the ZMI. This *manage* view shows what the system administrator can see in a personal folder. Figure 4 shows the user's view of his personal folder. Notice that the title field contains a search criterion (%u%). This shows one way that ZSQLCatalog can provide a SQL interface to data in Zope objects. Also note that the browser points to <http://xavante.cefetcampos.br:8080/nexedi/person>. The first part of this URL relates to the server and the port (in this case 8080) where Zope answers HTTP requests. The "/nexedi/person" part shows the actual folder hierarchy. Using templates for rendering the basic page structure and data from the objects in a person folder, Zope dynamically builds this final user view of the folder's contents.

Workflows, defined according to the business rules and by using the Zope workflow tool, regulate data and information sharing among components. The tool associates objects of a given type with workflows, each workflow representing a business rule and specifying the actions that are possible in any phase of the object's life cycle. In other words, Zope implements workflows through collaborative objects that know each other's location—its Zope path—and follow certain actions to implement business rules.

ERP5 ABSTRACT MODEL

ERP5 defines an abstract model for business management.

A clean abstraction layer helps consistently implement new or specialized business components. This abstract business model can represent management activities within a single company as well as commercial and management activities across a group of companies. This model's purpose is not to optimize production but rather to predict the consequences of management decisions. We based this model on five classes as Figure 1 shows:

- *Resource*. A resource describes an abstract resource in a business process (like an individual's skill, a currency, a raw material, or a product). Relations between nodes define bill of materials as well as prototypes.
- *Node*. Nodes can receive and send resources. They can relate to physical entities (such as a workshop that receives raw material, processes it, and sends it) or abstract entities (such as a bank account that can receive money).

Stocks are a type of node. Metanodes are nodes that contain other nodes. For instance, a company is a metanode, and a project is both a resource and a node.

- *Movement*. This class describes the movement of resources between two nodes at a given time and for a given duration. For example, a movement might send raw material from a stock to a workshop; another movement might send money from one account to another.
- *Path*. A path defines a way for a node to access a resource it might need. Prices and commercial profiles can attach to a path to define the default price for a given resource procured at a given maker. Paths can also define how a workshop obtains its resource from a stock. Paths have a start and end date, and can represent the assignment of an individual to a temporary project.
- *Item*. A physical instance of a resource is an item. A movement can expand into a series of traceable movements through items. Items also define how resources ship (such as by parcel or by listing the serial numbers of items in each container).

These five classes represent businesses ranging from chemical plants to services.

Transformations in the ERP5 model

Transformations represent a complex resource built from the transformation of multiple resources. Rather than following a hierarchical model, ERP5 uses a networked model based on the chemical transformation metaphor: $A + B \rightarrow C + D$.

To make it easier for users to define complex sets of trans-

formations, ERP5 can make prototype transformations where one transformation is equivalent to another except for a few differences. For example, transformation Y is like transformation X, but instead of producing resource R1, it produces resource R2, and instead of using material M1, it uses material M2. This creates new, complex resources that derive from existing complex resources.

Also, to implement choice of resource, equivalence resources allow defining one resource as any resource among a group of resources. Equivalence resources are useful to implement equivalence classes of resources for procurement from multiple sources. Transformations in the ERP5 model are instantiated into causalities of actual movements between nodes. The only causality that exists in a complex manufacturing process is from the definition of a transformation. Stocks (a type of node) have no causality: No one can ever say why a stock is below zero whenever multiple orders generate multiple outbound stock movements.

Movements and orders

The actual resource planning occurs in movements. Movements can include sub-movements that business rules generate. For example, a business rule could make the costing of a procurement result in a movement between source and stock nodes. Through causalities, different movements can also be associated with each other. Because movements have a beginning and an end (this is important, for example, to represent flows in a chemical process), the whole collection of movements with past dates and future dates represents the overall planning for the company. However, movements are low-level objects in ERP5. Users are not supposed to deal with movements except in rare cases (such as in accounting). Therefore, ERP5 gathers movements into deliveries. Deliveries are the documents that manage the company planning.

Path planning

Some companies, especially service companies, assign people (such as a consultant) to projects at certain periods of time or send people (such as a service technician) to customers at certain periods of time. ERP5 can represent these situations by using a temporary path with a given start and end date.

Capacity

Each node has a capacity that is measurable in two ways.

Stock capacity. The Stock Manager defines this capacity in terms of the maximum amount of a resource that a node can contain. A set of inequalities that the stock must satisfy



Resources

- ▶ **ERP5** (<http://www.erp5.org>): This site hosts source code, white papers, and news about ERP5.
- ▶ **Z Object Programming Environment (Zope) Community** (<http://www.zope.org>): The Zope open-source application server is tailored to provide dynamically generated content and is the basis for ERP5.
- ▶ **SyncML Protocol** (<http://www.openmobilealliance.org/syncml/>): This open standard seeks to drive data mobility by establishing a common language for communications among devices, applications, and networks.

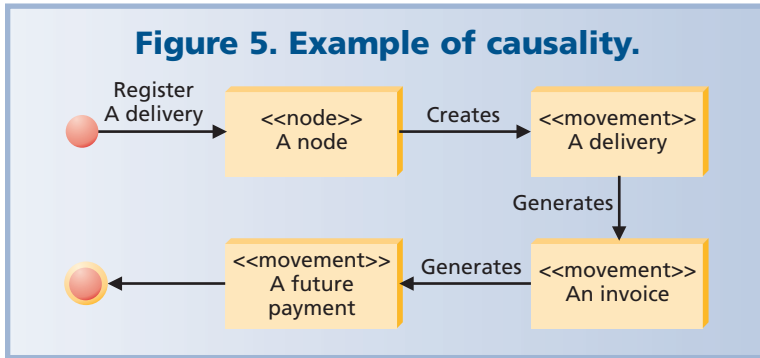
define the stock capacity. Each equation uses the amount of a given resource as a variable. The stock capacity is a parameter in these equations. To define a stock capacity inequality, we actually provide a set of points in N dimension. Each point defines a resource (or an “equivalence resource”) and a quantity. ERP5 then uses the convex hull of all points for capacity calculations. Because we consider that we only use convex areas, it is equivalent to provide a set of inequalities or a set of points. The planning is done by doing calculations on convex hulls or on cartesian products of convex hulls.

Production capacity. The production manager defines this capacity in terms of the maximum or minimum amount of a resource that the node can produce in a given period of time. A set of inequalities also defines production capacity. For example, a typical node can store at most one resource with a maximum quantity *MS* and produce at most one resource with a maximum production *MP* for each period of time. Planning at the node level requires accounting for movements that arrive or depart from that node. A movement that arrives at a node increases the level of stock. A movement that departs from a node decreases the level of stock. A movement that departs from a node and goes nowhere represents a consumption of resource, and a movement that departs from nowhere and goes to a node represents a production of resource. Calculating a node’s stock level requires looking at the history of movements, adding inbound amounts, and subtracting outbound amounts.

Simulating the future

The ERP5 abstract model simulates a company’s future, including orders, bills, shipments, and cash flow. Business rules handle the simulation process in ERP5. They transform a movement into another movement and generate deliveries of movements that are again transformed into new movements. For example, if a source on a node that has production capabilities creates a delivery of goods, then a manufacturing rule should generate all necessary movements (those for production, consumption, and deliv-

Figure 5. Example of causality.



ery). All generated movements then go into a causality that relates to the transformation.

A delivery of goods should generate an invoice movement, depending on the commercial conditions for the particular customer and the delivered goods.

Each invoice movement should generate a future-payment movement, depending on the commercial conditions, as Figure 5 illustrates.

Another example is the cost evaluation of a delivery. Each delivery between nodes should generate a movement to represent the cost of paying people to move objects across the workshop.

Another example is the salary rule. Each month a company should create a movement for each of its employees. That is, the movement should move an amount of a resource (cash) from the company's funds to the employee. This movement represents the payment of the employee's monthly salary. From this movement, a set of rules should generate movements to represent the payment of bonuses, insurance, and taxes.

From the previous examples, we can synthesize two generic concepts: Movements relate to each other by causalities; causalities relate to a rule.

There seems to be an origin to causality: One movement generates other movements. In other words, causality does not seem to be symmetric. This situation differs slightly from the more general symmetrical model first introduced for causalities.

Causality trees are much easier to implement than causality graphs. It also seems that trees, instead of graphs, can represent most causalities. For these reasons, we chose to implement causalities with trees in ERP5. Production planning in most companies usually combines pull approaches, where orders are the cause of build-order movements. This sort of triggering represents just-in-time, zero-stock management. In a push approach, inbound shipments cause build-order movements, an approach more appropriate for the food industry, for example.

Profiles

The whole simulation process sometimes must use predefined values to create new movements. Such values

might include prices, discounts, and payment options. ERP5 calls the comprehensive set of such values a profile object. Profile objects can attach to resources, nodes, paths, or even orders. A dynamic acquisition path—a specific URL—defined by business rules aggregates profile parameters.

ERP5 is now in production at the first European site, and development continues. The ERP5 Consortium—a group of independent organizations—organizes the ERP5 project. Consortium members are

users, developers, and IT companies that base commercial services and research efforts on ERP5. This consortium is in charge of coordinating research and development contributions, promoting the use of ERP5, and providing education and certification frameworks. Consortium members come from France, the Netherlands, and Brazil.

In the research arena, ERP5 members are developing advanced components for decision support, based on optimization, soft computing, and the data mining of object repositories. The first component, known as *Strategic Sourcing*, implements e-procurement. It uses fuzzy sets and integer programming to evaluate contract risk and to minimize the total cost of ownership for purchased items, while considering multiple concurrent sources.

ERP5 will also help increase the adoption of open source and free software technologies in small and medium-size enterprises, and will improve a market of interoperable services and solutions based on the ERP5 platform. We believe this will lead to increased innovation and transparency in the global ERP market by setting new industry standards. Research on management and organization sciences will find a reference platform in ERP5.

ERP5 will create a new channel for e-commerce applications, which often require tight integration with ERP software. Today, vendors of proprietary ERP systems are blocking the entry of new vendors into the market for e-commerce application software. If open source initiatives like ERP5 succeed, they will prevent these companies from having monopolistic control of future e-commerce applications. ■

Jean-Paul Smets-Solanes is the CEO of Nexedi SARL and the main developer of the ERP5 project. He is also heads the EuroLinux Alliance for a Free Information Infrastructure and is vice-president of the French Speaking Linux and Free Software Association. Contact him at jp@nexedi.com.

Rogério Atem de Carvalho is a teacher and researcher at the Federal Center for Technology Education of Campos-CEFET Campos, Brazil. Contact him at r.carvalho@computer.org.

For further information on this or any other computing topic, visit our Digital Library at <http://computer.org/publications/dlib>.