Ni, Qianfu and Yarlagadda, Prasad K.D.V. and Betts, Martin (2008) An Extensible Manufacturing Resource Model for Process Integration. *AIJST -Asian International Journal of Science and Technology : Production and Manufacturing* 1(4).

An Extensible Manufacturing Resource Model for Process Integration

Qianfu Ni[1], Prasad KDV Yarlagadda[1] and Martin Betts[1]

[1]Queensland University of Technology,
q2.ni@qut.edu.au, y.prasad@qut.edu.au, m.betts@qut.edu.au

ABSTRACT

Driven by industrial needs and enabled by process technology and information technology, enterprise integration is rapidly shifting from information integration to process integration to improve overall performance of enterprises. Traditional resource models are established based on the needs of individual applications. They cannot effectively serve process integration which needs resources to be represented in a unified, comprehensive and flexible way to meet the needs of various applications for different business processes. This paper looks into this issue and presents a configurable and extensible resource model which can be rapidly reconfigured and extended to serve for different applications. To achieve generality, the presented resource model is established from macro level and micro level. A semantic representation method is developed to improve the flexibility and extensibility of the model.

Keywords: resource modeling, process management and process integration.

1. INTRODUCTION

Since 1980s, various kinds of applications have been developed to assist enterprises to improve efficiency and productivity, such as computer-aided design (CAD), computer-aided process planning (CAPP), and computer-aided manufacturing (CAM) [1, 2]. These applications address requirements on some activities in enterprises and emphasize information creation with less concern about information management. These applications, working independently, result in duplication of the same piece of information and islands of automation to enterprise-wide integration [3]. Companies face the risk of program proliferation, of being burdened with a multitude of disconnected improvement efforts. The proliferation dissipates resources and creates confusion as people try to understand how each effort relates to the others. It also engenders harmful competition among specialists, with each group promoting its program as the most deserving of resources [4]. Recently, in order to enhance production efficiency and agility, the focus on business processes has gained common understanding in modern enterprises [5]. Some process-centric methodologies, such as business process re-engineering (BPR), virtual enterprise (VE), supply chain management (SCM), and collaborative product development (CPD) are investigated to enable business activities to be performed in a process context [6]. MIT Salon Management Review concluded that each performance improvement initiative needs to be positioned under a process management umbrella to boost operating results. Business process management in one form or another has become a feature in the language, if not the actions, of many organizations and is increasingly used by enterprises to improve the quality and efficiency of their administrative and production processes. It is used to manage ecommerce transactions and to rapidly and reliably deliver services to businesses and individual customers [7].

The implementation of business processes often involves integration of various applications. In enterprises, manufacturing resource information is widely shared by various business processes for different purposes at different stages of product development [8], as shown in Fig. 1. However, resource management capability, if it is really provided by traditional applications like CAD and CAM, is very limited. Usually, traditional applications only manage some of manufacturing resources for internal use without any attention to the needs of other applications. As most traditional applications are standalone and must be installed to individual computers, the coexistence of manufacturing resource information on different computers is inevitable. To support the integration of business processes, it is imperative to look at resources from a broader viewpoint and represent resources in a unified way. Resource information should be managed in a centralized database for better sharing. Such a manufacturing resource model should coherently represent various resources of enterprises so that each subsystem for different purposes and different people with different concerns can have consistent understandings of the same resources. To achieve integrity, a comprehensive and extensible data structure needs to be developed to accommodate as much information as possible to characterize every aspect of resources and relationships between different resources. By taking these concerns into consideration, this paper presents an extensible manufacturing resource model which is able to support integration of various business processes.

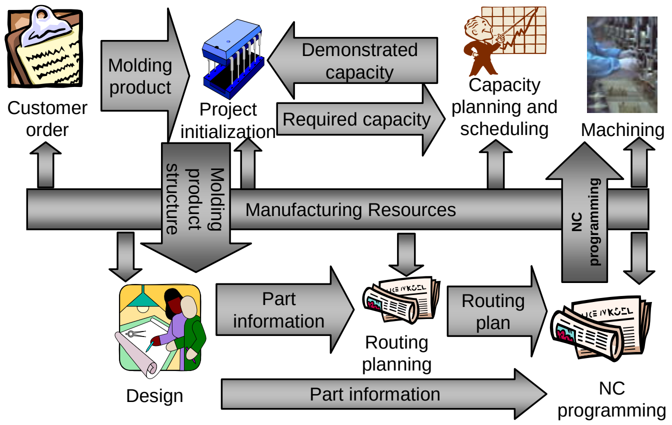


Fig. 1: Resource information sharing.

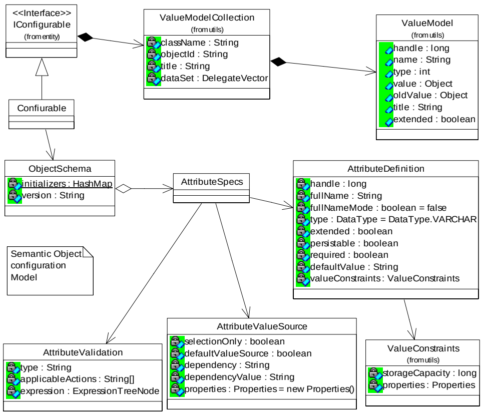


Fig: 2. Extensible object model.

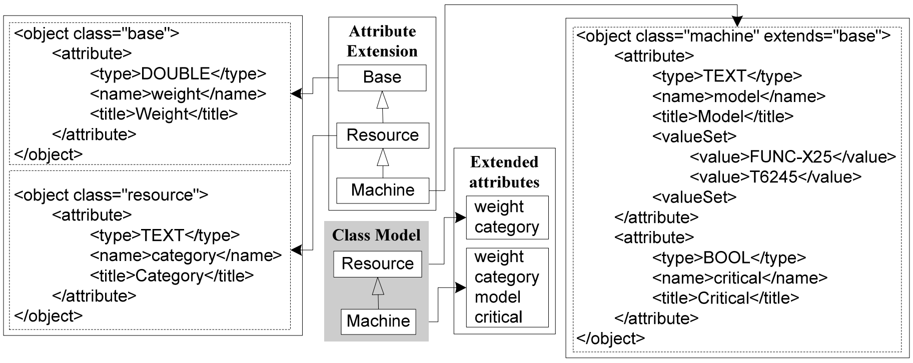
2. NEEDS OF EXTENSIBLE RESOURCE MODEL

Since the nineties, researchers started to work on the foundations of business process management system [10]. National Industry Information Infrastructure Protocols (NIIIP) intends to develop information infrastructures and supporting platforms for business process integration. Most of its earlier efforts are focused on reusing, extending and integrating various industry standards and enabling technologies, such as communications, object oriented technologies and information exchange technologies, such as Standard for the Exchange of Product Model Data (STEP). The European PRODNET II (Production Planning and Management in a Virtual Enterprise) focuses its research on integration of multiple standards or technologies, e.g. exchange of business messages, exchange of technical product data, and federated/distributed database [11]. IT infrastructure has been evolved from system centric to network centric to facilitate information integration. Now it is moving towards standard centric and will move further to process centric [12]. The standard centric stage can deemed as a stage leading to the process centric stage. In the standard centric infrastructure, standards play a very important role by providing business process with interoperability among different heterogeneous systems. These standards include CORBA (Common Object Request Broker Architecture), STEP and ebXML. CORBA addresses interoperability among objects in different language while STEP concentrates on product information exchange among different models used by CAD systems. EbXML provides a standard means for business process modeling, which covers requirement analysis, architecture and collaboration [13]. Traditionally, resource management is part of an enterprise application system, functionally organized for certain departments, e.g. computer aided process planning and production scheduling. Because such a resource management model is developed based on functional requirements raised by an application, they create information islands in an enterprise which leads to duplicate resource management efforts and information inconsistency problems.

To enable a resource model to fulfill the requirements of the entire business process, on the one hand, it should be generic enough to represent all resources. On the other hand, it should be able to be easily extended to represent and manage more detailed information and relationships to support a specific function and the extension should not cause impacts to the entire model and existing resource management functions. By achieving this goal, a resource model provides the following advantages: (1) Enabling incremental implementation of a process management solution at lower cost, (2) Enabling rapid implementation and deployment as less customization effort needed, and (3) Providing a flexible foundation to support business process optimization and redesign. This paper reports generic resource model, which can serve as a unified platform for application integration to support various business process in enterprises. The model consists of three interrelated sub-models: an extensible object model, a macro resource model and micro resource model. The extensible object model is an enabling technique to the extensible resource model. The macro model attends to provide a generic framework to represent common characteristics of resources while the micro model supports specific characterization of individual resources.

3. EXTENSIBILITY AND SEMANTIC REPRESENTATION

A flexible and extensible information model can enable the software system built on it be easily deployed to different companies with less customization effort. It also provides the flexibility to support changes to business strategies and operation management when needed. To maximize the extensibility of the developed resource model, a extensible object model has been developed as show in Fig. 2. The developed object model provides a foundation for defining extensible entities which can be extended using semantic representations. In the model, *EntityDefinition* aggregates *AttributeDefinition* to represent the definition of entities. *AttributeDefinition* is a class that represents attribute definitions by aggregating *AttributeSpecification*, *AttributeValidation* and *ValueSource*. *AttributeSpecification* is a class for holding core specifications of an attribute, such as attribute name, title and data type, which are compulsory to define an attribute. *AttributeSpecification* further aggregates *ValueConstraints*. *NumericConstraint*s, *TextConstraints* and *ObjectConstraints* are three subclasses of *ValueConstraints*, which respectively represent the value constraints of numerical type, text type and composite type. *AttributeValidation* and *AttributeValueSource* are classes to hold information defined in value validation and value source representations. The key techniques to enable attribute extension include 1) new attributes that can be defined after the system is developed; 2) an object should be capable of holding extended attributes; 3) values of the extended attributes validated as normal attributes at the runtime; and 4) extended attributes automatically maintained in the database. The foundation layer shown in Figure 4 offers information entities the capability to hold a set of extended attributes. In the model, *AttributeSpecs* is defined to describe attribute specs, such as attribute name, data type and display title, while *ObjectSchema* aggregates *AttributeSpecs* to represent the specifications of an object (a set of attributes). *IExtensible* is an interface that defines the fundamental capabilities required to support the attribute extension. Extensible is the default implementation of *IExtensible* to support the attribute extension. *AttributeValue* represents individual attribute values and *AttributeValues*, an aggregation of *AttributeValue*, is a collective data structure to hold extended attribute values of objects. As shown in the model, through aggregating *AttributeValues* by value, *Extensible* and its subclasses are capable of holding a set of extended attributes.



```
<object class="base">
        <attribute>
                <type>DOUBLE</type>
                <name>weight</name>
                <title>Weight</title>
        </attribute>
</object>
```

```
<object class="resource">
        <attribute>
                <type>TEXT</type>
                <name>category</name>
                <title>Category</title>
        </attribute>
</object>
```

```
<object class="machine" extends="base">
        <attribute>
                <type>TEXT</type>
                <name>model</name>
                <title>Model</title>
                <valueSet>
                        <value>FUNC-X25</value>
                        <value>T6245</value>
                <valueSet>
        </attribute>
        <attribute>
                <type>BOOL</type>
                <name>critical</name>
                <title>Critical</title>
        </attribute>
</object>
```

Fig. 3: Attribute inheritence.

To make attributes able to be extended at the implementation and deployment stage by the system implementer or end users, the extended attributes are defined using a descriptive language developed based on XML, called attribute declaration language (ADL). The ADL provides the capability to declare an attribute from four aspects: core, specialization, display, editing and validation as shown in Tab. 1. It also allows defining extended attributes by extending another set of extended attributes. In the example shown in Fig. 3, two extended attributes, model and critical, are defined for Machine. As the extended attributes of Machine are defined by extending the extended attribute set Base, the extended attribute weight is inherited by Machine. Besides, as Machine is the subclass of Resource, the extended attribute category of Resource is also inherited by Machine. Therefore, in total, Machine has four extended attributes. The keyword deprecated can be used to deprecate some attributes that are no longer useful. The deprecated attributes are made invisible to end users rather than physically removed from the system. By introducing keyword extended to indicate whether a declared attribute is an extended attribute or a built-in attribute, the ADL can be used to describe not only extended attributes but also modeled attributes to control their value editing, validation and attribute display. By doing so, the modeled attributes and extended attributes can be managed in the unified way. More important is that this approach makes the attribute validation able to be defined in the configuration file to prevent the logics from being hard coded and attribute display also can be flexibly changed via configuration without modifying the source code. The two approaches exist to store the extended attribute value. The one is to store AttributeValues as a binary data set, called BLOB in the database domain. Another approach is to store the extended attribute values as a long variable text by converting them to the XML-based text. When the object is retrieved from the database, the text is reversed back to AttributeValues. The first approach needs less time and effort because most computer languages directly support this data type. The disadvantage is that the low efficiency of the BLOB processing in databases can decrease the system performance. The second approach enables the system to achieve better performance. However, additional efforts are needed to do bidirectional conversion.

Type	Keyword	Explanation
Definition	name	Defines name of an extended attribute
	type	Defines data type of an extended attribute
	extended	Indicates its modeled or extended attribute
	defaultValue	Defines the default value of the attribute
Specialization	extends	Specifies parent schema
	deprecated	Indicates that an attribute is deprecated
Display	title	Defines a text as the display title of an attribute
	visible	Indicates whether the attribute is visible on screen
	prefix	Defines a text as the prefix of the attribute title
	suffix	Defines a text as the suffix of the attribute title
Editing	editable	Indicates whether the value is editable on screen
	required	Indicates whether the attribute is compulsory
	Suffix	Defines a text as the suffix of the attribute title
	valueSet	Define a set of possible values
Validation	maxValue	Defines a text as the display title of an attribute
	minValue	Defines a text as the prefix of the attribute title
	maxLength	Defines a text as the suffix of the attribute title
	decimalNum	Defines the number of digits after the decimal point
	uppercase	Automatically converts characters to uppercase
	lowercase	Automatically converts characters to lowercase
	spacePermitted	Indicates whether spaces in the attribute value is permitted

Tab. 1. The attribute declaration language

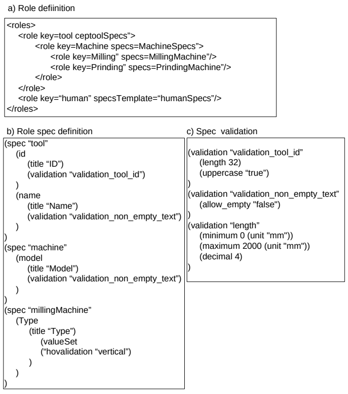
4. EXTENSIBLE RESOURCE MODEL

The resource information is utilized enterprise-wide by various business activities. It forms a running environment for the execution of different business processes consisting of various activities. To ensure that the activity is accomplished effectively and punctually, suitable resources should be selected by considering the resource capability and availability, and the resources selected have to be aligned with the resources used for the activities executed at the early stage. With the business operation management evolving from the function-based mode to the process centric mode, it is more imperative than ever before to develop a resource model capable of supporting the entire business processes and enabling optimized resource utilization [14].

By considering requirements of business process integration, the two-level enterprise resource model is developed as shown in Fig. 4. Based on the extensible foundation, a resource model is established as shown in Fig. 5. The model is divided into two levels to leverage the model generality and extensibility, i.e. macro level and micro level. At the macro level, the model is generic to represent all resources by addressing role, availability, capability and group while, at the micro level, the model can be extended to present further detailed information and relationships. At the macro level, *MacroResource* is a unified object modeled to represent all resources and *ResourceRole* plays a part in the resource categorization, such as machine and employee. As shown in the model, it is mandatory that a resource has one and only one role. Therefore, *MicroResource*, representing the association of *MacroResource* and *ResourceRole*, has one-to-one mapping relationship with *MacroResource*. The information associated with *MacroResource* is the generic information about the resource while *MicroResource* can be specialized to further characterize a resource. It can be seen from the model that any specialization to the MicroResource does not cause any impact to the macro level. As a result, this model is generic and extensible. As indicated in the model, ResourceRole is defined in the configuration file, as shown in Figure 4. In the configuration, name is the unique key to represent the role, and title defines information for screen display. The keyword schema specifies the key to find extended attributes configured to characterize the resource with the role. ResourceRole is self-associated, which enables a role to have sub-roles. In other words, the resource roles are organized in a tree structure. The extended attributes for a role can be inherited by its sub-roles and each sub-role also can have another set of extended attributes to further describe the resource. As shown in the model, the resource with the employee role has three extended attributes.

At the macro level, *MacroResource* is a unified object modeled to represent all resources and *ResourceRole* plays a part in the resource categorization, such as machine and employee. As shown in the model, it is mandatory that a resource has one and only one role. Therefore, *MicroResource*, representing the association of *MacroResource* and *ResourceRole*, has one-to-one mapping relationship with *MacroResource*. The information associated with

MacroResource is generic information about the resource while *MicroResource* can be specialized to further characterize a resource. It can be seen from the model that any specialization to the *MicroResource* does not cause any impact to the macro level. As a result, this model is generic and extensible. As indicated in the model, *ResourceRole* is defined in the configuration file. In the configuration, name is the unique key to represent the role, and title defines information for screen display. The keyword schema specifies the key to find extended attributes configured to characterize the resource with the role. *ResourceRole* is self-associated, which enables a role to have sub-roles. In other words, the resource roles are organized in the tree structure. By changing role configuration, this model can be tailored to represent resources for different companies. Resources can functions as groups, such as work centers and design teams, which are represented by *ResourceGroup* in the model. The model allows each resource to have a different role in the resource group. *IPurchasable* is an interface defined to represent the items that can be purchased form supplier, *IStockable* is an interface to represent the items which needs to be managed in the inventory and *IOutsourcible* represents the items that can be outsourced to partners.



a) Role defiinition

```
<roles>
    <role key=tool ceptoolSpecs”>
        <role key=Machine specs=MachineSpecs”>
            <role key=Milling” specs=MillingMachine”/>
            <role key=Prinding” specs=PrindingMachine”/>
        </role>
    </role>
    <role key=“human” specsTemplate=“humanSpecs”/>
</roles>
```

b) Role spec definition

```
(spec “tool”
    (id
        (title “ID”)
        (validation “validation_tool_id”)
    )
    (name
        (title “Name”)
        (validation “validation_non_empty_text”)
    )
)
(spec “machine”
    (model
        (title “Model”)
        (validation “validation_non_empty_text”)
    )
)
(spec “millingMachine”
    (Type
        (title “Type”)
            (valueSet
            (“hovalidation “vertical”)
        )
    )
)
```

c) Spec validation

```
(validation “validation_tool_id”
    (length 32)
    (uppercase “true”)
)
(validation “validation_non_empty_text”
    (allow_empty "false")
)
(validation “length”
    (minimum 0 (unit "mm"))
    (maximum 2000 (unit "mm"))
    (decimal 4)
)
```

Fig. 4: Role configuration.

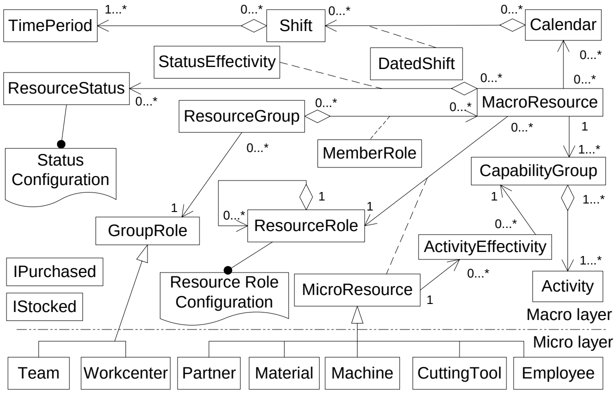


Fig. 5: Overview of the extensible resource model.

4.1 MACRO MODEL

At the macro level, *MacroResource* is a unified object modeled to represent all resources and *ResourceRole* plays a part in the resource categorization, such as machine and employee. As shown in the model, it is mandatory that a resource has one and only one role. *ResourceRole* is self-associated, which enables a role to have sub-roles. In other words, the resource roles are organized in the tree structure. Resources can functions as groups, such as work centers and design teams, which is represented by *ResourceGroup* in the model. The model allows each resource to have a different role in the resource group. *IPurchased* is an interface defined to represent the items that can be purchased and *IStocked* is an interface to represent the items which needs to be managed in the inventory. The information about resource availability is critical to generate accurate plans to ensure that activities can be performed in time without resource conflicts. If the activities are not well planned, some activities may compete for resources at a specific time, which can lead to the corruption of the business process. The resource availability is described from two aspects, namely capacity and status. The information about resource availability is critical to generate accurate plans to ensure that activities can be performed in time without resource conflicts. If the activities are not well planned, some activities may compete for resources at a specific time, which can lead to the corruption of the entire business process. The resource availability is described from two aspects, namely capacity and status.

4.1.1 Capacity Model

The resource capacity is key information to master planning and production scheduling. It is also utilized to manage task assignment in the workshops and the design department. In the model, resource capacity is represented by a set of shifts and calendars. The shift, consisting of a set of time periods, indicates time slots when a resource takes effective. For an instance, employees can be in office from 8:00am to 5:00pm, but they are not available in the period from 12:00pm to 1:00pm, which is lunch time. Therefore, the shift for the employees consists of two time periods: 8:00am to 12:00pm and 1:00pm to 5:00pm. The calendar organizes shifts based on date on a yearly basis. If a date has more than one shift linked, it means that the related resource works for multiple shifts on the date.

4.1.2 Status Model

The capacity model describes the resource availability over the timeline. However, at certain time, a resource might not be available because of unexpected accidents, such as machine breakdown and staff medical leave. The status model assists management of resource availability. The status information is also used for performance analysis, cost tracking and task management. The model uses *ResourceStatus* to represent the resource status and *StatusEffectivity* to record the time period when the related resource maintain their status. Different resources can have different meaningful status. In addition, different companies may manage different sets of status for the same resource. Taking a machine as an example, one company may only manage breakdown status in terms of availability while another company also manages monthly and yearly maintenance. To achieve the flexibility for companies to define the resource status according to their operation management strategies, the resource status is made configurable, as shown in Figure 8.

4.2 MICRO MODEL

The information associated with MacroResource is the common information about the resource. MicroResource can be specialized to further characterize a resource. It can be seen from the model that any specialization to the MicroResource won't cause any impact to the macro level. Therefore, the model is generic and extensible. This paper will deal only with three micro models to demonstrate the concept of extensibility.

4.2.1 Material Model

Material information is commonly shared by design, process planning and NC programming. Materials are characterized by physical properties and chemical properties. Designers should consider material physical properties, such as tensile strength, and select proper materials for parts to ensure the performance and life of parts and products. From the concurrent engineering viewpoint, material machining performance should also be considered. In processing planning, material chemical properties must be considered in the selection of cutting tools. In the machining process, high chemical affinity between the materials of a cutting tool and a part can cause sticking, even chemical changes, due to high temperature. This can dramatically decrease the quality of machined surfaces. The cutting tools made of tenacious materials are recommended to cut hard and brittle materials to reduce the chipping possibility of cutting edges. For tenacious part materials, cutting tools made of hard materials should be selected to achieve a high material removal rate by using high cutting speeds. In nature, cutting tool selection is a knowledge-intensive decision-making process. However, no sound theoretic model exists to represent the relationships between cutting tool materials and part materials to guide the decision-making. This paper proposes two relationships between part materials and tool materials to represent this knowledge, which are compatible relationship and repellent relationship. The former is the one which indicates that tool materials are highly recommended for associated part materials. The later implies that tool materials are not proper for associated part materials. Material information is shared by many activities in different processes, such as design, planning and purchasing. A unified material model sharable to all related activities can maximize the material information sharing and facilitate streamlining of the relevant processes to improve working efficiency.

Selecting proper materials for parts is the designers' responsibility. In the selection of part materials, the critical factors concerned by designers are the material physical properties, such as tensile strength, key to the performance and life of parts and products. Normally, in a design drawing, the designer can partially or fully specify the material specifications, such as material code, heat treatment by the material manufacturer, also called raw status, shape and size. If only the material code is specified, the planner has to decide the blank shape and size for the part while material planning by taking into consideration dimension accuracy, surface roughness and material machining performance. The material plans, the result of the material planning, provide information about when and what material stocks as well as standard parts need to be ready for use. They are the official documents to support the purchasers to prepare and issue purchasing orders. In the market, the shapes and sizes of material stocks are standardized. Non-standard stocks cost more and need longer delivery time as they only can be specifically ordered from the manufacturer. It can be seen that the designer and planners select materials based on the part specification and performance while purchasers look at the materials from the market viewpoint.

Apart from the part materials, tool materials also need to be managed to support manufacturability evaluation and the process planning. From the routing planner' viewpoint, while selecting cutting tools for machining operations in the process planning, the chemical properties of materials receives more attention. During the cutting, the high chemical affinity between a cutting tool material and a part material can cause sticking and chemical change due to the high temperature. As a result, it dramatically decreases the surface quality. The cutting tools made of tenacious materials are recommended to cut hard and brittle materials to effectively reduce the chipping possibility of the cutting edge. For the part materials with high tenacity, cutting tools made of hard materials need to be selected to achieve the higher material removal rate by using high cutting speeds.

At the micro level, the material model shown in Figure 7 is specialized by incorporating the concerns discussed above. Material, a subclass of *MicroResource*, is defined to represent materials and, further, *PartMaterial* and *ToolMaterial* are specialized from *Material* to characterize part materials and cutting tool materials respectively.

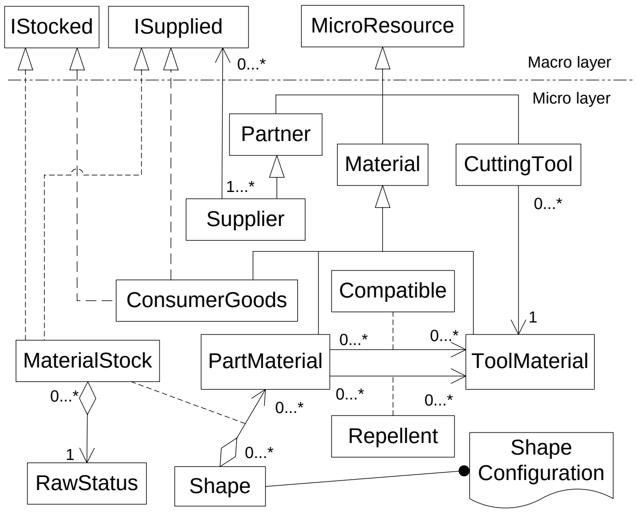


Fig. 6: Micro material model

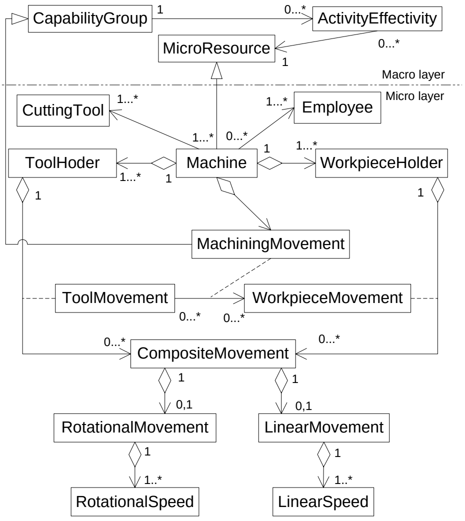


Fig. 7: Micro machine model

MaterialStock is the association of *PartMaterial* and *Shape*, representing various sizes and shapes of the material stock. *PartMaterial*, an abstract concept without shape and size, provides information about the chemical and physical properties. *MaterialStock* defines what shapes and sizes of materials can be purchased from different suppliers which offer different prices and delivery times. Shape can be configured based on the market standard. RawStatus represents the heat treatment done by manufactures. Cutting tool selection is a knowledge intensive process [15]. However, no sound theoretic model exists to represent the relationships between the cutting tool materials and the part materials to guide the decision-making. This paper utilizes two relationships, Compatible and Repellent, to represents best and worst matches between the cutting tool materials and the part materials. The relationships can be built by senior process planners and used to assist junior planners and designers to perform tasks.

4.3.2 Machine Tool Model

Machines are one of the critical resources in the manufacturing companies and the machine information is needed by designers, planners, financial staff and maintenance staff to accomplish their tasks. For the purpose of manufacturability evaluation, designers need to know the machine capability while designing. At the micro level, the machine model, as shown in Fig. 7, focuses on characterizing the machine capability.

The machining capability can be described by three elements: operation, feature and precision that can be achieved. Usually, one type of machine can do multiple types of operations. In addition, for a machine with multiple spindles or multiple tool holders, the capabilities depend on the spindles or the tool holders. It is necessary to develop a comprehensive model to represent the machine capabilities in a unified way.

Theoretically, a feature is formed by the relative movement between a cutting tool and a work piece. For instance, rotation of a work piece and linear movement of a tool form a cylindrical surface. As mentioned above, some machines can have multiple spindles and/or tool holders. Therefore, it is only meaningful to discuss machine capabilities based on the pair of tool and work piece movements. Some machines, such as lathes, use a spindle as the work piece holder and others, such as milling machines, use the spindle as a tool holder while a working table is used as a work piece holder. In order for the capability representation to be generic to different machines, six abstract movements in the machine model illustrated in Fig. 7 are used to assist the definition of the machine capability. In the model, *LinearMovement* represents a natural linear movement characterized by *LinearSpeed*, which can be a set of values in case of discrete speed gearing or a speed range for continuous speed gearing. Similarly, *RotationalMovement* represents a rotational movement described by *RotationalSpeed* with a set of values or a speed range depending on gearing. *CompositeMovement* represents the natural movement composite of the linear movement and the rotational movement. As the model shows, *CompositeMovement* can have zero or one rotational movement and zero or one linear movement. In total, four possible types of *CompositeMovement* exist: 1) the *CompositeMovement* movement which has neither *LinearMovement* nor *RotationalMovement* linked, which means 'static'; 2) the *CompositeMovement* movement which only has a *LinearMovement* linked, which means a linear movement; 3) the *CompositeMovement* movement which only has a *RotationalMovement* linked, which means a rotation movement; and 4) the *CompositeMovement* movement which links to Linear *Movement* and a *Rotational Movement* linked, which simultaneously does both linear and rotational movements.

As shown in the model, a machine can have one or more tool holders and workpiece holders. Each tool holder and workpiece holder can link to zero or more *CompositeMovements*. Jointly, *ToolMovement* and *WorkpieceMovement* form a *MachiningMovement*. As shown in the model, *MachiningMovement* is specialized from *CapabilityGroup* modeled at the macro level. As a result, each *MachiningMovement* instance can link to a set of *ActivityEffectivity* instances to characterize the capabilities of the movement.

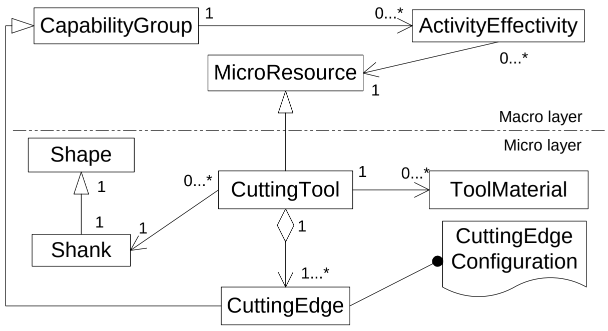


Fig. 8: Micro cutting tool model

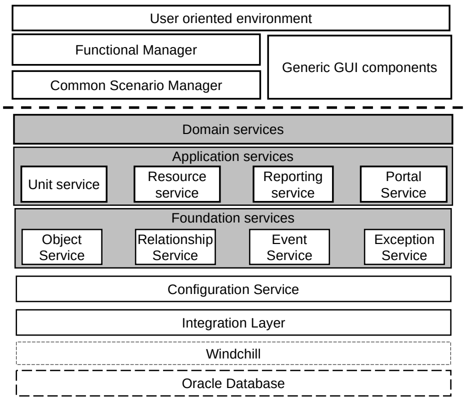


Fig. 9 System architecture

4.3.3 Cutting Tool Model

Effective cutting tool information available at design stage can assist designers to make better decisions to reduce proliferation of cutting tools, which is the problem that nearly 70% of manufacturing companies are struggling to tackle [15]. To leverage the concurrent engineering, the collaborative environment makes the cutting tool information available to designers so that designers can evaluate feature manufacturability based on cutting tools to make better decisions and make the existing cutting tools able to be reused as much as possible. As a result, time for tool making or purchasing can be reduced and the production cost also can be lowered down. To address this issue, the micro cutting tool model is developed, as shown in Fig. 8. The model attends to represent the complicated tools consisting of several cutting edges. For the manufacturability evaluation, three major kinds of information, i.e. tool shank, tool material and cutting edges, are concerned. The tool shank, an interface to the tool holder of machines, is represented by *Shank*, a subclass of Shape defined at the macro level. Similar to machine tools, the tool capability is represented as the combination of operation, feature and precision. The model can represent the cutting tools which have multiple cutting edges. The tool capability is managed based on the cutting edge, represented by *CuttingEdge*. It should be aware that *CuttingEdge* is also a subclass of *CapabilityGroup*. Therefore, the capabilities of each cutting edge can be specialized. Based on these three micro resource models, the collaborative environment provides the function to evaluate the feature manufacturability according to the machine capability and cutting tool capability with consideration of match between part materials and tool materials

5. PROTOTYPE

In general, enterprise integration involves process analysis and redesign, integration platform selection and integration development to deliver a tailored solution to meet specific requirements [16]. The integration platform affects the solution performance and determines the effort and cost needed for integration. Two major concerns are stressed in the platform selection: 1) the architecture of the platform provides the powerful foundation for business integration; 2) the platform offers more ‘green areas’ to enable clean and neat integration. A powerful foundation provides fundamental functions, such as security management and data persistence, so that the development efforts can be minimized and the integration can focus on the business level. A “green area” is referred to where the functionality required is almost fully provided and can be directly used in the integrated environment or does not exist at all. On the contrary, the situation where the functionality is provided and can only partially meet the requirements is called the “grey area”. According to our experience, compromise is often needed while enhancing the existing functions according to the new requirements. The green area offers a great freedom to incorporate new concepts into an integrated solution. In other words, a powerful and green platform enables delivery of an optimized integration solution with less customization efforts and short implementation lifecycle at lower cost. Windchill, a PDM system from PTC, is selected as the platform to implement the integration because Windchill can fully fulfill the design management requirement with a little customization. At the same time, it offers a spacious green room to design and implement the resource management and the production integration. Windchill has a built-in workflow engine and offers a GUI (graphic user interface) based environment for end users to define, track and monitor workflows. Winchill is a good platform for process integration.

5.1 Architecture

Leveraged by Windchill, the multi-tier and service-oriented architecture shown in Fig. 9 is developed. The architecture can be divided into the service layer and the representation layer. The service layer provides the functions to process business logics. The representation layer defines the structure of GUI components and the communication mechanism among the components.

5.1.1 Integration Layer

At the service layer, in order to minimize the potential impacts caused by the changes of Windchill and to prevent the system being tightly coupled with Windchill, an integration layer is introduced. This layer consists of our foundation classes, relationship patterns and service patterns, extensible mechanism, configurable mechanism and authorization management mechanism.

5.2 Service Layer

All the services are organized into the three layers: configuration services, foundation services, application service and domain services.

1) Configuration service: The configuration service is designed to parse and manage various kinds of configurations discussed above, such as object, relationship and resource status configuration.

2) Foundation services: The foundation services include object service, relationship service, event service and exception service, which are the common services to leverage enterprise functions development. The object service is capable of instantiating extensible objects based on the object schema configured in ADL and the relationship service provides functions to manage relationship and do relationship navigation based on the relationship configuration. The event service provides functions to manage and dispatch events. Other services can be registered as event publishers or subscribers. In the service, all events are organized in a hierarchical structure to enable a publisher or a subscriber to work on a branch of events rather than all events. For example, under the root can be quotation event, customer order event and under the quotation event can be creation, modification, deletion and cost estimation. The creation event may further broken down into quotation creation and quoted item creation. The service enables a subscriber to subscribe some branches, such as the quotation event or the creation event. The lower level branches are automatically included in the upper level branch. For example, when the quotation event is subscribed, all the events under the quotation event will be dispatched to the subscriber.

The exception service provides the common functions to process the server side exceptions. Primarily, it provides a common function to capture exceptions and log the exceptions into corresponding log files for analysis.

3) Application services: The application services consist of the unit service, resource service, reporting service and portal service, which are application-level functions to support the domain services. Different industries may have different sets of common units and a specific company may only deal with several currencies in their business though many currencies exist in the world. To enable end users to define the units which are commonly used in the company, the unit configuration service is developed for end users to tailor units based on their requirements.

The resources service is developed based on the resource model discussed above. It mainly provides two types of functions. One is to provide GUI based environment for end users to define and manage the resources they have. Another is to provide the services for domain services or other applications to access resource information.

Reports are significant and critical deliverables from enterprise applications to end users. They provide summarized and concise information for end users to effectively capture resource status, track job progresses and analyze profit profiles etc. Besides, they are also key documents for managers to communicate with customers, do planning and make decisions. Many customization efforts are needed to develop specific reporting functions for the targeted company while implementing an enterprise application. The reporting service developed attends to provide a unified function to generate different reports by using Microsoft Excel spreadsheets as templates and XML-based configurations. The spreadsheets is used to define the look and feel of reports and the configurations are used to define information to be rendered into reports, information position and text styles, such as bold, italic and size. The service also provides the function to manage different reports in a centralized way.

The employees with different roles are authorized to use different functions and view different kinds of information. An employee may play multiple roles. In this case, the functions offered to all the roles need to be combined to present to the employee. This service is developed with the capability of dynamically organizing and presenting system functions based on the roles played by the current users.

4) Domain services: The domain services are the services related to different activities, such as quotation processing, project management. They need to be modeled and designed by corresponding exporters.

5. 3 Service Layer

The presentation layer is modeled and designed based on the component-based technology by using the model-view-control (MVC) principal. On this layer, a scenario manager represents the control to process the logics on the client side. It can communicate with the various services to acquire information and to trigger information processing on the server side. The GUIs are responsible for information presentation based on the unified data model, such as PropertyModel discussed above, serve as the interfaces between the manager and end users. The GUIs forwards the user operations to the scenario manager for processing. The scenario manager also notifies the GUIs to update components status when necessary. Firstly, the common scenarios, such as creation of new objects, update of objects, deletion of objects or query objects, are abstracted and developed into the common scenario manager. The common manager mainly focuses on establishing the mechanism to trigger various functions, such as information validation before and after user operations and authorization check before user operation and information presentation, and defining the protocol, such as event dispatching or methods invocation, to facilitate communication with the GUIs. On the top of the common scenario manager, the functional scenario managers, such as the quotation manager and the customer order manager, are developed to realize the functions. A set of commonly GUI components are developed to speed up the GUI development. Various functions are organized together according to the roles of the current user to form user specific working environment to improve the working efficiency.

5. CONCLUSION

To address the needs of integrated manufacturing, an extensible manufacturing resource model has been modelled. A flexible and extensible information model presented here enables the software system built on it to be easily deployed to different companies with less customization effort. By considering requirements of business process integration, the two-level enterprise resource model is developed. Also based on the extensible foundation, a resource model is also established. The model is divided into two levels to leverage the model generality and extensibility, i.e. macro level and micro level. At the macro level, the model is generic to represent all resources by addressing role, availability, capability and group while, at the micro level, the model can be extended to present further detailed information and relationships. A prototype has been developed as a proof-of-concept system to demonstrate the developed system architecture and models.

6. REFERENCES

[1] Yunshun Fan et al 1999. Enterprise wide application integration platform for CIMS implementation. J0urnal of Integlligent Manufacturing Vol. 10:587-601.

[2] Gunasekaran, H. B. Marri, etc 2000. Design and implementation of Computer Integrated Manufacturing in Small and Medium-Sized Enterprise: A Case Study. Int. J. of Advanced Mfg. Technology, No. 16:46-54.

[3] Cheolhan Kim et al 1993. An Object-Oriented Information Modeling Methodology for Manufacturing Information Systems. Computer Industry Engineering Vol. 24 No. 3:337-353.

[4] Michael Hammer 2002, Process Management and the Future of Six Sigma, IEEE Engineering Management Review Vol. 30 No. 4:56-63.

[5] Hongmei Gou et al 2003. A Framework for virtual enterprise operation management, Computers in Industry Vol. 50:333-352

[6] Colin Armistead et al 1999. Strategic Business Process Management for Enterprise Effectiveness. Long Range Planning Vol. 32 No. 1:96-106

[7] Daniela Grigori et al 2004. Business Process Intelligence. Computers in Industry No. 53:321-343

[8] Liu, C., Wang, X., and He, Y., Research on manufacturing resource modeling based on the O–O method, Journal of Materials Processing Technology, Vol. 139, No. 40-43, 2003, pp 40-43.

[9] Brandon.P.S. and Betts.M.(editors), Integrated Construction Information, E & FN Spon, UK, 1995.

[10] Wil M. P. van der Aalst 2002 Making Work Flow: On the application of Petri Nets to Business Process Management ICATPN 2002:1-22.

[11] L.M. Camarinha-Matos & H. Afsarmanesh 1999, Infrastructures for Virtual Enterprises—Networking Industrial Enterprises. Dordrecht: Kluwer Academic Publishers

[12] Simon Liu, A Practical Framework for Distributing IT Infrastructure. IT Professional No. 7:14-20

[13] Ann Lindsay et al 2003. Business processes—attempts to find a definition. Information and Software Technology, Vol. 45:1015–1019 2003

[14] Mezgar, I., Kovacs, G.-L., and Paganelli, P., Co-operative production planning for small- and medium-sized enterprises, International Journal of Production Economics, Vol. 64, No., 2000, pp 37-48.

[15] R.P. Baker and P.G. Maropoulos, “An architecture for the vertical integration of tooling considerations from design to process planning”, Robotics and Computer Integrated Manufacturing, 16 (2000) 121-131

[16] Doumeingts G, Ducq Y, Vallespir B, Kleinhans S, Production management and enterprise modelling. Computers in Industry 2000; 42(2-3):245-263.