EXTENSIBLE MODEL NETWORK REPRESENTATION SYSTEM FOR PROCESS PLANNING

TECHNICAL FIELD OF THE INVENTION

This invention relates in general to the field of process planning. More particularly, the invention relates to an extensible model system architecture for integrated material and capacity planning, and integrated factory and distribution planning.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced



BACKGROUND OF THE INVENTION

Manufacturers and distributors commit to time critical production and delivery of goods as a regular part of their business. Often, the manufacturing and distribution process is complex, having many different material and capacity constraints that simultaneously affect the implementation of the process. Due to the complexity of the processes, many different software tools have been developed to help manufacturers and distributors plan. Most such tools only address narrow parts of the overall planning process. Several unrelated tools may be required to fully plan a process.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

SUMMARY OF THE INVENTION

The present invention provides an extensible model architecture for process planning that eliminates or reduces problems with prior systems.

More particularly, in one embodiment, the present invention provides a computer software system for modeling a process capability on a computer. The computer software system comprises an operation model type for defining a plurality of operation models. Each operation model represents an activity that can be performed by a process. A resource model type is for defining a plurality of resource models. Each resource model represents capacity available for use in performing an activity and rules for allocating capacity to the activity. A buffer model type is for defining a plurality of buffer models. Each buffer model represents rules for controlling a flow of material between activities. The operation model type, buffer model type, and resource model type each comprise a plurality of fields defining attributes. The plurality of fields includes a plurality of extension selector fields that each allow a user to specify one of a plurality of extensions that augment a model with additional fields and semantics. The model type specifies a base set of fields and semantics which includes fields that select extensions that specify additional fields and semantics that can be added to a particular model. The presence of a field in a model is dependent upon a value of another field in the model. Defined operation models, buffer models, and resource models are stored as nodes in an interrelated process network model. The process network model is formed by a plurality of operation models each specifying buffer models from which material is consumed and buffer models to which material is supplied and specifying resource models having capacity used in performing the activity specified by the operation model.

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

In this manner, both material and capacity usage are simultaneously represented along with timing constraints between activities.

According to the present invention, to model a plan for a process, a plan network model is constructed upon the process network model. An operation-plan model type is used to define a plurality of operation-plan models. Each operation-plan model represents a plan for performing an activity represented by an operation model. A resource-plan model type is used to define a plurality of resource-plan models. Each resource-plan model represents a planned usage of capacity represented by a resource model. A buffer-plan model type is used to define a plurality of buffer-plan models. Each buffer-plan model represents a planned flow of material represented by a buffer model. A plan network model is formed by a plurality of operation-plan models each specifying resource-plan models planned to be used in performing the activity specified by the operation-plan. The nodes of the plan network model are built upon and refer to the nodes of the process network model. According to the present invention, the process network model represents possibilities of what can be done, and the plan network model represents that which is planned to be done.

According to one aspect of the present invention, each model type has one or more extension points that may be used to extend the basic model type in order to support the information needed to create the user defined models. For each extension point, a user may select an appropriate extension to model a particular aspect of the user's system.

According to another aspect of the present invention, several elements of a user's process model may share data in a hierarchical fashion referred to as families. A child model inherits all of the data of the

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

parent model unless it is specifically overridden by a designation in the child model. A software system implementing the teachings of the present invention may provide for date effectivity of models using the families feature.

It is a technical advantage of the present invention to use extensions to define the elements of the process or system being modeled because it reduces the amount of data that needs to be stored in a memory device without reducing modeling power. According to the teachings of the present invention, data for each element of the modeled process is stored at a node of a user model or process network model. The data stored at a node comprises only the data for the fields corresponding to the extensions that are selected by the user. The system does not store a zero quantity for each field not chosen by the user. Therefore, the use of extensions reduces the size requirement for a memory device in a system implementing the present invention.

The software system of the present invention can be, but not necessarily, implemented using object oriented programming. Use of object oriented programming supports simultaneous use of multiple types of models for the different elements of the planning problem.

This Sheet Has Been Replaced



BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings in which like reference numbers indicate like features and wherein:

FIGURE 1 is a graphical representation of a portion of a process network model implementing an extensible model architecture for use in planning a process according to the teachings of the present invention;

FIGURE 2 is a graphical representation of a plan network model implementing an extensible model architecture for use in planning a process according to the teachings of the present invention; and

FIGURE 3 is a graphical representation of a family of resource models in a process network model implementing inheritance;

FIGURE 4 is a graphical representation of a family of resource models in a process network model implementing date effectivity.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced



DETAILED DESCRIPTION OF THE INVENTION

In one embodiment, the present invention provides a software system for modeling various processes. For example, the software system may model a manufacturing process used to produce a particular item or product. The software system may also be used to model a product distribution channel, a supply chain, or an order fulfillment process.

According to the present invention, a process is modeled using three primary model types: an operation model type, a buffer model type, and a resource model type. A plan for that process is similarly modeled with three associated primary model types: the operation-plan, buffer-plan, and resource-plan. A user uses a model type as a template to create a model. For example, a buffer model type is used to create a model of a buffer. Each of these user defined models is stored as a node in a process network model of a user's process.

The model types used by the system are extensible in that each model type may have one or more extension points that allow a user to customize the model to represent the user's process. For each extension point, a user may choose from numerous extensions the one that is best suited to define a particular aspect of the user's system or process. Each model type has a list of predefined extensions from which the user may choose. Each extension defines fields that extend the data and meaning associated with a model type. These extensions and fields define the way in which the model types interact. Each user defined model is defined by a model type, one extension for each extension point of the model type, and data for each field of the model type and extensions. Possible features of each of the model types are discussed in detail below.

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

The Operation Model Type

The operation model type may be used to create operation models which represent activities that can be performed. As used herein, "operation model type" means a general model for activities that may be modeled by the software. "Operation node" and "operation model" are used interchangeably to represent a specific operation defined by a user using the operation model type. This convention is also used with respect to buffers and resources.

An operation model type models a process, activity, or action that transforms or moves items resulting in a flow into, out of or between buffers. Operations may require resources with specific skills. Those resources model the capacity to perform operations. Buffers model the flow of items that result from operations. Operations model the activity.

Activities may include transportation, storage, picking, receiving, testing, assembling, packaging, manufacturing, designing, setup, maintenance, and other actions. An operation can use any number of resources, with different run times, and with staggered start and end dates. Thus, a single activity or a whole series of activities can be modeled with a single operation. This is important for having sufficient flexibility to model diverse activities in different manufacturers, even within a single supply chain.

The modeling needs of these varied activities can be very different. To support those variations, the operation model can be extended via a 'process' extension. Different extensions each can add different fields to the operation model. A simple process extension may add a single field such as a fixed amount of time that the operation runs. A more complex process extension may add a fixed time and a per-unit time that

is computed proportional to the number of units that operation performed. A much more complex process extension may add a field that is a list of other operations which are performed in sequence in order to perform this operation.

Thus, unlike traditional manufacturing software that separate routings and operations, the present invention allows a routing to be modeled as simply a particular kind of operation - an operation that consists of other operations that are run in sequence. The relationships between those sub-operations can be different depending upon the chosen extension. So, an operation can model a simple routing (a sequence of operations allowed to spread), a flow routing (a sequence of operations which must flow into one another), a set of operations that can be run at the same time, alternates (a set of alternate operations), and other combinations.

Further, since routings are modeled as operations, they can be put in other operations. For example, a simple routing operation may consist of five operations. One of those five may be an alternates operation, which consists of three possible operations. One of those possibilities may itself by a simple routing operation consisting of three operations. Thus, depending upon which alternate is chosen, there may be five or seven operations to perform.

An operation can also model simply the bill-of-materials for an item (operations that just model the transform of items, but do not model the capacity required to do that). Similarly, an operation can model simply the bill-of-distribution for an item (operations that just model the movement of items between buffers, but do not model the capacity required to do so). Other operations can then combine the bill-of-materials or bill-of-distribution operations with operations that

properly consume capacity. Such separation is common in older manufacturing databases.

Some operation specifications can define multiple operations. For example, some manufacturing software defines operations such that resources can be loaded just during portions of operation time. For ease of interfacing such packages or databases, process extensions can be provided with the identical specification. That process generates a routing-like operation containing auto-generated sub-operations that contain the different phased resource loadings.

An advantage of the operations model type design is simplicity and consistency. By providing a simple building block that is extensible and can be flexibly combined with others, a great deal of modeling capability is provided without complicating the common simple cases. In this way, the critical operations and resources can be modeled in adequate detail while the non-critical models remain suitably inexpensive.

A user may select an extension from the operation model type to, for example, define a transportation operation model. Alternately, a user may choose an extension from the operation model type that creates a smelting operation model for smelting iron ore to produce steel in a smelter. In essence, every act that needs to be performed by a process is modeled using the operation model type.

An example of an extension for an operation model is a run extension. A simple run extension may consist of one field: the fixed amount of time that the operation model is to run. For example, a particular operation that runs for 2 hours may be modeled by that extension. Alternatively, a more complicated run extension may have three fields: such as a load-run-unload. This extension may define a predetermined, constant time for loading material, followed by a run time which is a factor that

is multiplied by the quantity of material that is being processed, followed by an unload run time which is constant. The load, run, and unload phases may each require use of different resources.

Operation models represent activities that consume items, produce items, and/or utilize capacity. Such operation models define the activities and precedence (timing) constraints in a process.

In manufacturing, a sequence of operations is referred to as a routing. A routing is represented as an Operation model with a routing extension which defines the behavior of that operation to be a sequence of other operations. Similarly, the specification of alternate resources, alternate operations, and alternate routings can be done with an operation model and an alternates extension which defines the behavior of that operation to be "perform one of these other operations". Additionally, an operation model may maintain constraints between successive operation models such as perish times (i.e. the maximum time between operations).

The Operation-Plan Model Type

An operation-plan model type may be used to create operation-plan models which represent the plans to perform activities that are represented by operation models. The operation-plan model refers to an operation model that defines the allowed behavior. The operation-plan model represents a choice among the allowable behaviors, or even a choice of a disallowed behavior (an infeasible plan). For example, the operation-plan model would include the dates during which the operation will be performed. If different resources may be chosen to perform the activity, the operation model represents those options, whereas the operation-plan model represents a choice of the resource planned to be used.

Extensions selected for the operation model can cause corresponding extensions to be selected for a operation-plan model that refers to the operation model. Those extensions add fields and semantics that correspond with the semantics of the operation model. For example, an extension may add minimum and maximum limits for a value in fields of the operation model, and a corresponding extension may add the planned value as a field of the operation-plan model.

The Resource Model Type

A resource model type may be used to create resource models which represent aspects of a process that have a predetermined capacity. For example, a resource model may represent laborers used in a manufacturing process. A resource model may represent a physical object such as a truck, a forklift, an oven, or any other object used in implementing the process that has a predetermined capacity limitation (such as space, electricity, or money).

The resource model type may provide various extensions from which a user may choose to define an actual capacity constraint on its process. For example, a simple extension may only have a single field setting a maximum capacity constraint. Such a resource model may, for example, represent a truck which has a maximum carrying capacity of 10 tons. Alternatively, a more complex extension may be chosen. For example, the extension may have fields relating to both a maximum capacity and a minimum capacity. Extensions for the resource model type may also have fields for constraints such as the number of set up times allowed in a particular time period or the number of hours that may be devoted to setting up the particular resource.

A resource model may have an efficiency extension that defines in one of a number of different ways how the

efficiency of that resource varies over time. The efficiency may be defined as a fixed value, as a value that ramps up over time (a learning curve), or as a value that changes seasonally as specified in a calendar.

A resource model may have a variability extension that defines how the plan should be padded to compensate for the variability of that resource. A resource model may also have a maintenance extension that defines when maintenance is required. One extension may compute based on time since the last maintenance, the load-time during it, a weighted sum of load-times, the number of setups or the setup time, and other such times. It may also determine separately both major and minor maintenance needs.

A resource model may have a size extension that defines the size of the resource and defines how the size is consumed by the various operations. For example, a truck trailer may have a size of up to ten tons and up to twelve feet high, eight feet wide, and 26 feet long. The resource would also know how to compute the weight and the physical dimensions of the operations that use the resource. A resource model may also have a load policy extension that defines the rules for loading that resource (e.g. no overlapping operations, and no operations longer than three hours on Mondays).

Operation models may reserve capacity of predetermined resource models. For example, a user may define a transportation operation model for transporting an item. The transportation operation may reserve a truck resource model to carry out the act of transporting the item.

An operation model may require more than one resource at a time. For example, in a milling process, a particular operation model may specify that it requires a milling machine resource model and a labor resource model. Additionally, an operation may reserve alternate

resource models. Finally, a resource model may specify inter-operation constraints such as sequence-dependent set up times or combination constraints. For example, a resource model for an oven may specify that all operations that use the same temperature and are less in size than available space can use a particular oven at the same time.

The resource model type can be used to model and manage a storage space. A storage space may represent, for example, a warehouse, a store room, a refrigerator, or floor space in a factory.

The Resource-Plan Model Type

A resource-plan model type may be used to create resource-plan models which represent the planned availability and usage of the capacity represented by resource models. The resource-plan model refers to a resource model which defines the allowed behavior. The resource-plan model represents a choice among the allowable behaviors, or even a choice of a disallowed behavior (an infeasible plan). For example, the resource-plan model would include the dates during which the resource will run overtime. The resource model represents the options for performing overtime, whereas the operation-plan model represents a choice of the overtime being planned.

Extensions selected for the resource model can cause corresponding extensions to be selected for any resource-plan model that refers to the resource model. Those extensions add fields and semantics that correspond with the semantics of the resource model. For example, an extension may add minimum and maximum limits for a value in fields of the resource model, and a corresponding extension may add the planned value as a field of the resource-plan model.

The Buffer Model Type

The buffer model type is used to create buffer models which manage items consumed and/or produced in the modeled process. In other words, a buffer model represents the management of a particular item. The buffer model type may have extensions that represent constraints such as minimum and maximum inventory levels. Additionally, the buffer model type may have an extension that defines ordering policies such as lot-for-lot, or fixed size batches. A buffer model type extension may also require a "safety stock" that must be on hand at all times in case of emergency.

A buffer model manages the flow of materials to and from an operation model. A buffer model may provide sheet metal to a press for stamping the sheet metal into a particular shape. As part of its definition, an operation model will point to the buffer models that it consumes items from and to the buffer models that it supplies items to. It is noted that a defined buffer model may not have an actual inventory associated with it. Rather, the defined buffer model may simply represent an item flow control point. Alternatively, a buffer model may maintain an inventory of items sitting between operations.

The Buffer-Plan Model Type

A buffer-plan model type may be used to create buffer-plan models which represent the planned availability and usage of the capacity represented by buffer models. The buffer-plan model refers to a buffer model which defines the allowed behavior. The buffer-plan model represents a choice among the allowable behaviors, or even a choice of a disallowed behavior (an infeasible plan). For example, the buffer-plan model would include the dates during which the buffer will run overtime. The buffer model represents the options for

performing overtime, whereas the operation-plan model represents a choice of the overtime being planned.

Extensions selected for the buffer model can cause corresponding extensions to be selected for any buffer-plan model that refers to buffer model. Those extensions add fields and semantics that correspond with the semantics of the buffer model. For example, an extension may and minimum and maximum limits for a value in fields of the buffer model, and a corresponding extension may add the planned value as a field of the buffer-plan model.

Process Network Model

FIGURE 1 is a graphical representation of a portion of a process network model indicated generally at 10 according to the teachings of the present invention. A process network model is a set of user defined models which may be stored in a computer memory. The stored nodes define, for example, the user's factory which is to be used to manage a particular manufacturing process. A user defines the process network model 10 of FIGURE 1 by selecting model types and the appropriate extensions of these model types, as described above, to form models of the elements of the users' process. The nodes in process network model 10 may be connected together in many different ways to achieve different results. Process network model 10 represents all of the possible interconnections between the user defined models. For example, process network model 10 of FIGURE 1, in part, consists of four operation models, 11, 12, 13, 14 and 15, and four buffer models, 16, 18, 20 and 22. Material may flow through various paths in process network model 10.

Operation model 11 of FIGURE 1 may represent the act of stamping a piece of sheet metal into one of many different predetermined shapes as modeled by operation models 12, 13, 14 and 15. Additionally, buffer models 16

and 18 may supply two different types of sheet metal. The portion of process network model 10 shown in FIGURE 1 shows several alternative material flow paths.

For example, sheet metal may be provided by buffer 16 to operation 13 and then to buffer 20. Alternatively, sheet metal may be provided by buffer 16 to operation 14 and then buffer 22. Sheet metal may be provided by buffer 18 to operation 14 and then to buffer 20. Alternatively, sheet metal may be provided by buffer 18 to operation 15 and then buffer 22.

Process network model 10 represents the possible routes that material may flow in a process to be modeled. The software system of the present invention uses the information in a process network model to create plans for implementing a particular material flow path to model a process. The interconnected nodes of the process network model used to represent a particular plan may be referred to as a plan network model or plan.

The software system of the present invention may create a plan network model as a result of several different events. In general, a plan network model provides definitions as to quantities and timing for elements of the model. For example, the inventory level for an item managed by a particular buffer model may drop below an allowable level. In response, the process network model may create a plan network model for replenishing the inventory of that buffer model. Alternatively, a plan may be created using the process network model in response to customer orders.

The software system of the present invention may create a plan network model as follows. For each activity in a user's process, the software system creates an operation plan model from an operation-plan model type using extensions from the corresponding operation model. The system provides the operation plan model with the start and end dates, alternate selections, and other

THIS SHEET HAS BEEN REPLACED

information needed to form a plan for a particular execution of that operation.

For each resource model that is used by an operation plan, the software system creates a resource plan model from a resource-plan model type and the extensions corresponding to the resource model. The system provides the resource plan model with all of the capacity reservations placed on it by the operation plan models.

For each buffer model from which items are consumed or to which items are supplied by an operation plan, the software system creates a buffer plan model from a buffer-plan model type and the extensions of a corresponding buffer model. The system provides the buffer plan model with inventory levels, ordering policies and any other information needed to plan the management of the flow of items in the process. The various plan models interact to create a schedule of events to implement the modeled process to produce a selected quantity of output.

Plan Network Model

FIGURE 2 is a graphical representation of a plan network model for producing, for example, bicycle fenders in a predetermined process. For purposes of FIGURE 2, it is assumed that a user has previously defined a site network having all of the operation-plan, buffer-plan, and resource-plan models necessary to model the user's physical facilities and process requirements. In response to a condition for creating a plan as identified above, each operation-plan, buffer-plan, and resource-plan model required for use in the plan creates plan models. The flow of material through the manufacturing process for creating bicycle tires as modeled by the software system of the present invention is described in detail below.

This Sheet Has Been Replaced



The storage space holding sheet metal for creating bicycle fenders is represented by resource-plan model 32. At operation-plan model 33, sheet metal from resource-plan model 32 is made available in buffer-plan model 34. Buffer-plan model 34 manages the flow of sheet metal. At operation-plan model 36, the sheet metal is stamped into a predetermined shape for the fender. Stamp operation-plan model 36 requires two resource-plan models. First, resource-plan model 38 comprises an industrial press with a die selected to press the sheet metal into a predetermined shape. It is noted that the die may be modeled as a separate resource. A second resource-plan model 40 for operation-plan model 36 may be a laborer that operates press resource-plan model 38.

Buffer-plan model 42 manages the flow of pressed sheet metal from operation-plan model 36 to operation-plan model 44. At operation-plan model 44, the pressed sheet metal is painted. Painting operation-plan model 44 may use resource-plan models 46 and 48 such as a vat of paint and a laborer.

The final step in preparing the fender is to dry the paint. Buffer-plan model 50 manages the flow of material from painting operation-plan model 44 to drying operation-plan model 52. Operation-plan model 52 utilizes a resource-plan model 54 that may comprise an oven or other appropriate heating element to dry the paint. Heating resource-plan model 54 is operated by resource-plan model 56 which may represent a laborer. The dried fenders are pulled from buffer-plan model 60 and stored in a storage space modeled by resource-plan model 58 by operation-plan model 57.

Families

Nodes in the process and plan network models may be interrelated with common elements in a hierarchical order referred to as families. Each node can have only one

family. In a family relationship, the children nodes inherit all the extension and field values of their family nodes unless they are specifically overridden by the child. Any changes to fields of a parent model are effective for the child models that do not override those fields.

FIGURE 3 shows an example of the use of families for a resource model. Resource models 70, 72 and 74 are all ovens. Resource model 70 is the family. Resource models 72 and 74 are the children. Both resource model 72 and resource model 74 have all of the same extensions as resource model 70. Model 72 inherits all values from model 70. Model 74 inherits all values except it has a larger size and higher efficiency.

Use of families allows planning at different levels. In this example, the information stored at the node for resource model 70 may be used to perform aggregate planning. Whereas the information stored at the nodes for resource models 72 and 74 may be used to perform more detailed plans as to what particular ovens are to be used for particular operations.

Date Effectivity

Families may be used to provide an efficient manner in which to update changes to a model. Date effective changes to a model generally affect only a small number of the total fields. It would be tedious and error-prone to have to maintain, for example, many independent copies of the same information in date effective variations. The principle of inheritance in families may be used to efficiently implement date effective changes to a model. As shown in FIGURE 3, gas oven resource model 78 is only valid until a specified day, DATE. Beginning on DATE, electric oven resource 80 must be implemented. A date effective field of resource 78 dictates that it may only

be used prior to the DATE. After the DATE, only resource model 80 may be used.

Although the present invention has been described in detail, it should be understood that various changes, substitutions and alternations can be made hereto without departing from the spirit and scope of the invention as defined by the appended claims.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

WHAT IS CLAIMED IS:

1. A computer software system for modeling a process capability on a computer, the computer software system comprising:

an operation model type for defining a plurality of operation models, each operation model representing an activity that can be performed by a process;

a resource model type for defining a plurality of resource models, each resource model representing capacity available for use in performing an activity and rules for allocating capacity to the activity; and

a buffer model type for defining a plurality of buffer models, each buffer model representing rules for controlling a flow of material between activities;

wherein the operation model type, buffer model type, and resource model type each comprise a plurality of fields defining attributes, the plurality of fields including a plurality of extension selector fields that allow a user to specify one of a plurality of optional extensions incorporating additional fields and semantics in addition to fields specified by the model type into each model selecting the optional extension; and

wherein defined operation models, buffer models, and resource models are stored as nodes in an interrelated process network model, the process network model formed by a plurality of operation models each specifying buffer models from which material is consumed and buffer models to which material is supplied and specifying resource models having capacity used in performing the activity specified by the operation model, such that both material and capacity usage are simultaneously represented along with timing constraints between activities.

This Sheet Has Been Replaced

2. The computer software system of Claim 1, further comprising:

an operation-plan model type for defining a plurality of operation-plan models, each operation-plan model representing a planned activity to be performed during a particular period in order to achieve a particular purpose;

a resource-plan model type for defining a plurality of resource-plan models, each resource-plan model representing an amount of available capacity and planned usage of the available capacity by operation-plan models; and

a buffer-plan model type for defining a plurality of buffer-plan models, each buffer-plan model representing planned flow of material that controlled by the buffer-plan model as a result of planned operation-plan models;

wherein the operation-plan model types, buffer-plan model types, and resource-plan model types each comprise a plurality of fields that define attributes, including a plurality of extension selector fields corresponding to operation, buffer, and resource model type fields and specifying corresponding extension of a plurality of optional extensions which incorporate additional fields and semantics into each model selecting the optional extension in addition to those specified by the model type; and

wherein the defined operation models, buffer models, and resource models are stored as nodes in an interrelated process-plan network model, the process-plan network model formed by operation-plan models specifying buffer-plan models from which material is consumed and buffer-plan models to which material is supplied and specifying resource-plan models having capacity planned to be used in order to perform the activity specified by the operation-plan model, such that both material and capacity usage are simultaneously represented and

addressed along with operation timing constraints in creating and managing a plan for the process.

3. The computer software system of Claim 1, wherein the operation model type has associated with one of the extension selectors a plurality of extensions that define behavior as a sequence of other operations, with varying rules on how the sequence may be ordered or may spread out in time.

4. The computer software system of Claim 2, wherein:

the operation model type has associated with one of the extension selectors a plurality of extensions that define behavior as a sequence of other operations, with varying rules on how the sequence may be ordered or may spread out in time; and

the operation-plan model type has a plurality of extensions each corresponding to each of the extensions of the operation model type, which represent extension-specific choices for a particular plan for the process.

5. The computer software system of Claim 1, wherein the operation model type has associated with one of the extension selectors a plurality of extensions that define behavior as one of several other operation models, with varying rules on how a single one of those operation models is chosen to be performed.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

6. The computer software system of Claim 2, wherein:

the operation model type has associated with one of the extension selectors a plurality of extensions that define behavior as one of several other operation models, with varying rules on how a single one of those operation models is chosen to be performed; and

the operation-plan model type has a plurality of extensions each corresponding to each of the extensions of the operation model type, which represent extension-specific choices for a particular plan for the process.

7. The computer software system of Claim 1, wherein the operation model type has associated with one of the extension selectors a plurality of extensions that define behavior as a set of other operations each having individual precedence constraints, with varying rules on how the set of operations is planned.

8. The computer software system of Claim 2, wherein:

the operation model type has associated with one of the extension selectors a plurality of extensions that define behavior as a set of other operations each having individual precedence constraints, with varying rules on how the set of operations is planned; and

the operation-plan model type has a plurality of extensions each corresponding to each of the extensions of the operation model type, which represent extension-specific choices for a particular plan for the process.

THIS SHEET HAS BEEN REPLACED

9. The computer software system of Claim 1, wherein the operation model type includes a plurality of extensions that define behavior as a sequence of other operations, as one of several other operation models, and as a set of other operations each having individual precedence constraints, such that a complex hierarchy of operation models can be formed in order to represent complex processes with alternatives, sequences, and precedence constraints.

10. The computer software system of Claim 2, wherein:

the operation model type includes a plurality of extensions that define behavior as a sequence of other operations, as one of several other operation models, and as a set of other operations each having individual precedence constraints, such that a complex hierarchy of operation models can be formed in order to represent complex processes with alternatives, sequences, and precedence constraints;

the operation-plan model type has a plurality of extensions each corresponding to each of the extensions of the operation model type, which represent extension-specific choices for a particular plan for the process; and

a plan for such a complex process can be represented with all decisions to be made in order to define a particular plan.

11. The computer software system of Claim 1, wherein the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantics that each describe how an efficiency of the resource model changes over time.

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

12. The computer software system of Claim 2, wherein:

the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantic that each describe how an efficiency of the resource model changes over time; and

the resource-plan model type has a plurality of extensions each corresponding to each of the extensions of the resource model type, which represent extension-specific choices for a particular plan for the process.

13. The computer software system of Claim 1, wherein the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantics that each describe a maximum size of operation models that simultaneously use a resource model.

14. The computer software system of Claim 2, wherein:

the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantics that each describe a maximum size of operation models that simultaneously use a resource model; and

the resource-plan model type has a plurality of extensions corresponding to each of the extensions of the resource model type, which represent extension-specific choices for a particular plan for the process.

15. The computer software system of Claim 1, wherein the resource model type has associated with one of the extension selectors a plurality of extensions that define fields and semantics that each describe when maintenance operations should be planned.

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

16. The computer software system of Claim 2, wherein:

the resource model type has associated with one of the extension selectors a plurality of extensions that define fields and semantics that each describe when maintenance operations should be planned; and

the resource-plan model type has a plurality of extensions corresponding to each of the extensions of the resource model type, which represent extension-specific choices for a particular plan for the process.

17. The computer software system of Claim 1, wherein the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantics that each describe how operation plans should be padded to compensate for variability of a resource model.

18. The computer software system of Claim 2, wherein:

the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantics that each describe how operation plans should be padded to compensate for variability of a resource model; and

the resource-plan model type has a plurality of extensions corresponding to each of the extensions of the resource model type, which represent extension-specific choices for a particular plan for the process.

This Sheet Has Been Replaced

19. The computer software system of Claim 1, wherein:

the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantics that each describe how an efficiency of the resource model changes over time, describe a maximum size of operation models that simultaneously use a resource model, describe when maintenance operations should be planned, and describe how operation plans should be padded to compensate for variability of a resource model;

such that a wide variety of resource models can be represented in one network without greatly increasing complexity of each resource model requiring only a small subset of the overall software system modeling capabilities.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

20. The computer software system of Claim 2, wherein:

the resource model type has associated with one of the extension selectors a plurality of extensions defining fields and semantics that each describe how an efficiency of the resource model changes over time, describe a maximum size of operation models that simultaneously use a resource model, describe when maintenance operations should be planned, and describe how operation plans should be padded to compensate for variability of a resource model; and

the resource-plan model type has a plurality of extensions each corresponding to each of the extensions of the resource model type, which represent extension-specific choices for a particular plan for the process;

such that a wide variety of resource models and a plan for each of the resource models containing all decisions regarding capacity can be represented in one network without greatly increasing complexity of each resource model requiring only a small subset of the overall software system modeling capabilities.

21. The computer software system of Claim 1, wherein the buffer model type has an operation that defines how material managed by a buffer is stored, and the specified operation model can be extended to specify different scrap factors.

22. The computer software system of Claim 2, wherein the buffer model type has an operation that defines how material managed by a buffer is stored, and the buffer-plan models create operation-plan models to store the material managed by the buffer.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

23. The computer software system of Claim 1, wherein the buffer model type has an operation that defines how material is received into a buffer and how the material is picked from the buffer.

24. The computer software system of Claim 2, wherein:

the buffer model type has an operation that defines how material is received into a buffer and how material is picked from the buffer; and

the buffer-plan models create operation-plan models to receive each lot of material and pick each lot consumed.

25. The computer software system of Claim 1, wherein:

the buffer model type has an operation that defines how material managed by a buffer is stored, and the specified operation model can be extended to specify different scrap factors and has an operation that defines how material is received into a buffer and how the material is picked from the buffer;

such that a wide variety of resource models can be represented in one network without greatly increasing complexity of each resource model requiring only a small subset of the overall software system modeling capabilities; and

such that a complex hierarchy of operation models can be formed to represent complex processes with many alternatives, sequences, precedence constraints.

This Sheet Has Been Replaced
THIS SHEET HAS BEEN REPLACED

26. The software system of Claim 2, wherein:

the buffer model type has an operation that defines how material managed by a buffer is stored, and the specified operation model can be extended to specify different scrap factors and has an operation that defines how material is received into a buffer and how the material is picked from the buffer; and

the buffer-plan models create operation-plan models to receive each lot of material and pick each lot consumed and to store the material that is managed by the buffer at all times;

such that a wide variety of resource models and a plan for each of those resource models containing all decisions regarding capacity can be represented in one network without greatly increasing complexity of each resource model requiring only a small subset of the overall software system modeling capabilities; and

such that a complex hierarchy of operation models can be formed to represent complex processes with many alternatives, sequences, precedence constraints, and a plan for such a complex process can be represented with all the decisions to be made in order to define a particular plan.

27. The computer software system of Claim 25, used to model a manufacturing process wherein: operation models are used to represent traditional manufacturing operations, routings, bills-of-material, and storage; resource models are used to represent machines, tools, fixtures, and labor; and buffer models are used to represent SKUS, bins, buffers, or inventories.

THIS SHEET HAS BEEN REPLACED
This Sheet Has Been Replaced

28. The computer software system of Claim 26, used to model and plan a manufacturing process wherein:

operation models are used to represent traditional manufacturing operations, routings, bills-of-material, and storage; resource models are used to represent machines, tools, fixtures, and labor; and buffer models are used to represent SKUS, bins, buffers, or inventories; and

operation-plans, resource-plans and buffer-plans are used to represent corresponding plans for those entities.

29. The computer software system of Claim 25, used to model a distribution network wherein: operation models are used to represent transport, loading, unloading, storage, receiving, picking, palletizing, and packaging activities and traditional bills-of-distribution; resource models represent docks, trucks, pallets, forklifts, labor, and storage facilities; buffer models represent SKUS, bins, and inventories.

30. The computer software system of Claim 26, used to model a distribution network wherein:

operation models are used to represent transport, loading, unloading, storage, receiving, picking, palletizing, and packaging activities and traditional bills-of-distribution; resource models represent docks, trucks, pallets, forklifts, labor, and storage facilities; and buffer models represent SKUS, bins, and inventories; and

operation-plans, resource-plans, and buffer-plans are used to represent corresponding plans for those entities.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

31. The computer software system of Claim 25, used to model a manufacturing process wherein:

operation models are used to represent traditional manufacturing operations, routings, bills-of-material, and storage; resource models are used to represent machines, tools, fixtures, and labor; and buffer models are used to represent SKUS, bins, buffers, or inventories; and

operation models are used to represent transport, loading, unloading, storage, receiving, picking, palletizing, and packaging activities and traditional bills-of-distribution; resource models represent docks, trucks, pallets, forklifts, labor, and storage facilities; buffer models represent SKUS, bins, and inventories;

such that a full supply chain can be modeled.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

32. The computer software system of Claim 26, used to model and plan a manufacturing process wherein:

operation models are used to represent traditional manufacturing operations, routings, bills-of-material, and storage; resource models are used to represent machines, tools, fixtures, and labor; and buffer models are used to represent SKUS, bins, buffers, or inventories;

operation models are used to represent transport, loading, unloading, storage, receiving, picking, palletizing, and packaging activities and traditional bills-of-distribution; resource models represent docks, trucks, pallets, forklifts, labor, and storage facilities; and buffer models represent SKUS, bins, and inventories; and

operation-plans, resource-plans and buffer-plans are used to represent corresponding plans for those entities;

such that a full supply chain can be modeled and planned.

33. The computer software system of Claim 25 used to model project plans wherein: operation models are used to represent requirements gathering, design, testing, prototyping, tooling, and production; resource models are used to represent the labor, machines, tools, and testers; and buffer models represent various milestones to be achieved in order to move from one activity to another.

This Sheet Has Been Replaced

34. The computer software system of Claim 26 used to model project plans wherein:

operation models are used to represent requirements gathering, design, testing, prototyping, tooling, and production; resource models are used to represent the labor, machines, tools, and testers; and buffer models represent various milestones to be achieved in order to move from one activity to another; and

operation-plans, resource-plans, and buffer-plans are used to represent corresponding plans for those entities.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

35. The computer software system of Claim 25, used to model a manufacturing process wherein:

operation models are used to represent traditional manufacturing operations, routings, bills-of-material, and storage; resource models are used to represent machines, tools, fixtures, and labor; and buffer models are used to represent SKUS, bins, buffers, or inventories;

operation models are used to represent transport, loading, unloading, storage, receiving, picking, palletizing, and packaging activities and traditional bills-of-distribution; resource models represent docks, trucks, pallets, forklifts, labor, and storage facilities; buffer models represent SKUS, bins, and inventories; and

operation models are used to represent requirements gathering, design, testing, prototyping, tooling, and production; resource models are used to represent the labor, machines, tools, and testers; and buffer models represent various milestones to be achieved in order to move from one activity to another;

such that a full supply chain can be modeled and a supply chain and design activities involved in introducing new products and phasing out old products together with on-going production and delivery of existing products can be represented.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

36. The computer software system of Claim 26, used to model and plan a manufacturing process wherein:

operation models are used to represent traditional manufacturing operations, routings, bills-of-material, and storage; resource models are used to represent machines, tools, fixtures, and labor; and buffer models are used to represent SKUS, bins, buffers, or inventories;

operation models are used to represent transport, loading, unloading, storage, receiving, picking, palletizing, and packaging activities and traditional bills-of-distribution; resource models represent docks, trucks, pallets, forklifts, labor, and storage facilities; and buffer models represent SKUS, bins, and inventories; and

operation models are used to represent requirements gathering, design, testing, prototyping, tooling, and production; resource models are used to represent the labor, machines, tools, and testers; and buffer models represent various milestones to be achieved in order to move from one activity to another; and

operation-plans, resource-plans, and buffer-plans are used to represent corresponding plans for those entities;

such that a full supply chain can be modeled and planned and such that a supply chain and design activities involved in introducing new products and phasing out old products together with on-going production and delivery of existing products can be represented and planned.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

37. The computer software system of Claim 1, wherein each operation model may specify another operation, each resource model may specify another resource, and each buffer model may specify another buffer as its family and may specify a number of fields as inherited from that family, thereby allowing the fields to be changed in the family by a user, and result in changes to all models that inherit that field.

38. The computer software system of Claim 37, wherein each operation model, resource model, and buffer model that specifies a family can also specify a range of dates during which the model is effective such that outside of the range the model is modeled as if the model does not exist.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

39. A computer system for executing a computer software system for modeling a process capability, the computer system comprising:

a data storage device operable to store data relating to a model of process capability;

an execution memory operable to store data representing a computer software system; and

a processor coupled to the data storage device and to the execution memory, the processor operable to execute the computer software system, wherein the computer software system comprises;

an operation model type for defining a plurality of operation models, each operation model representing an activity that can be performed by a process;

a resource model type for defining a plurality of resource models, each resource model representing capacity available for use in performing an activity and rules for allocating capacity to the activity; and

a buffer model type for defining a plurality of buffer models, each buffer model representing rules for controlling a flow of material between activities;

wherein the operation model type, buffer model type, and resource model type each comprise a plurality of fields defining attributes, the plurality of fields including a plurality of extension selector fields that allow a user to specify one of a plurality of optional extensions incorporating additional fields and semantics in addition to fields specified by the model type into each model selecting the optional extension; and

wherein defined operation models, buffer models, and resource models are stored as nodes in an interrelated process network model, the process network model formed by a plurality of operation models each specifying buffer models from which material is consumed and buffer models to which material is supplied and

This Sheet Has Been Replaced
THIS SHEET HAS BEEN REPLACED

specifying resource models having capacity used in performing the activity specified by the operation model, such that both material and capacity usage are simultaneously represented along with timing constraints between activities.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced

40. The computer system of Claim 39, wherein the computer software system further comprises:

an operation-plan model type for defining a plurality of operation-plan models, each operation-plan model representing a planned activity to be performed during a particular period in order to achieve a particular purpose;

a resource-plan model type for defining a plurality of resource-plan models, each resource-plan model representing an amount of available capacity and planned usage of the available capacity by operation-plan models; and

a buffer-plan model type for defining a plurality of buffer-plan models, each buffer-plan model representing planned flow of material that controlled by the buffer-plan model as a result of planned operation-plan models;

wherein the operation-plan model types, buffer-plan model types, and resource-plan model types each comprise a plurality of fields that define attributes, including a plurality of extension selector fields corresponding to operation, buffer, and resource model type fields and specifying corresponding extension of a plurality of optional extensions which incorporate additional fields and semantics into each model selecting the optional extension in addition to those specified by the model type; and

wherein the defined operation models, buffer models, and resource models are stored as nodes in an interrelated process-plan network model, the process-plan network model formed by operation-plan models specifying buffer-plan models from which material is consumed and buffer-plan models to which material is supplied and specifying resource-plan models having capacity planned to be used in order to perform the activity specified by the operation-plan model, such that both material and

This Sheet Has Been Replaced
THIS SHEET HAS BEEN REPLACED

capacity usage are simultaneously represented and addressed along with operation timing constraints in creating and managing a plan for the process.

THIS SHEET HAS BEEN REPLACED

This Sheet Has Been Replaced