Int J Adv Manuf Technol (2010) 46:737–755
DOI 10.1007/s00170-009-2148-y

ORIGINAL ARTICLE

An approach for generating a tasks schedule model in web-based virtual manufacturing system of screw threads

Janus S. Liang

Received: 13 January 2009 / Accepted: 25 May 2009 / Published online: 9 June 2009

Abstract In this paper, several web-based interfaces for user interaction and a task-oriented decision approach are proposed to build a tasks schedule model and then display the manufacturing process in a virtual environment, which is created by a geometric virtual-reality-based visualization technology for screw threads generation. The proposed tasks schedule model consists of four types of objects: virtual component, state manager, transfer operator, and flow controller. The virtual component has a geometric model with kinematics and their attributes. To control the geometric model, a component controller which models the logical aspects of a component is used. The component controller should be able to implement component-level orders by operating the geometric model. For the fidelity of the tasks schedule model, a transfer operator has a set of component-level command imitating the physical mechanism of a transfer. As a result, more accurate simulation results can be expected. The flow controller makes decisions on friable transfers based on decision parameters, which are maintained by the state manager. To have better structure and easier implementation, a virtual manufacturing platform can be modeled in a hierarchical and modular manner as an integrated system consisting of a product design suite, a web interface module, and a visualization module. Meanwhile, it provides a solution of learning of manufacturing sequences, cost effective, platform independent, and sharing visualized information over the internet for virtual manufacturing. Finally, the tasks schedule model has been implemented with an example in screw threads generation.

J. S. Liang (✉)
Department of Vehicle Engineering,
Yung-Ta Institute of Technology and Commerce,
316 Chunshan Road, Linlo,
Ping Tung 909, Taiwan
e-mail: janus@mail.ytit.edu.tw

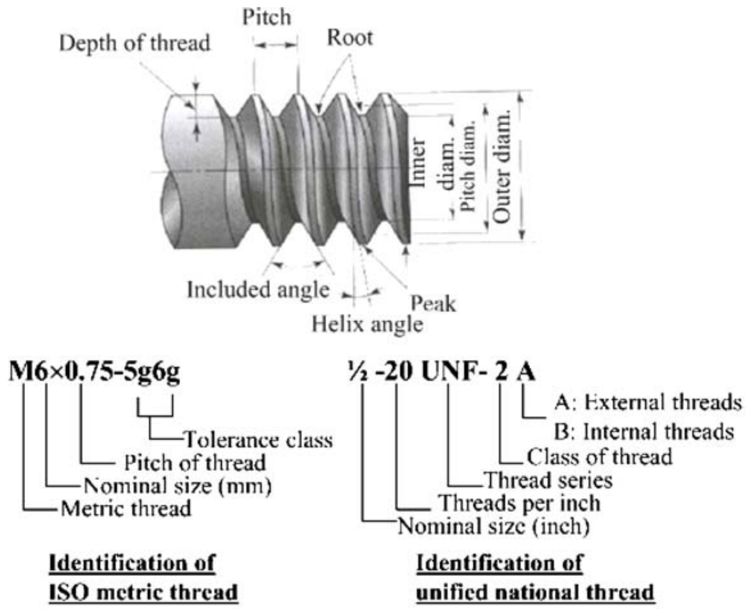
Keywords Virtual manufacturing environment · Screw threads generation · Tasks schedule model

1 Introduction

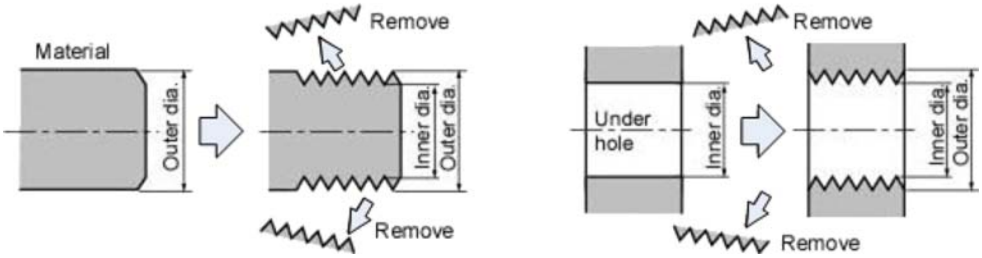
Tasks schedule plays a key role in manufacturing to transform design specifications into essential information for final product. It decides processes and sequence of product creation. Meanwhile, a tasks schedule generated for a product being designed can be used to either determine manufacturing feasibility or calculate reliable time and cost in completing the intended task. If it not feasible to be used in an existing manufacturing environment, the re-planning or re-designing of a product has to be executed. This paper proposes a method and takes screw threads generation an example to build a tasks schedule model in virtual environment.

Screw threads are among the most important machine elements, as you can note by observing screws, bolts, and other threaded components in machines and various products. A screw thread may be defined as a ridge of uniform cross-section that follows a spiral or helical path on the outside or inside of a cylindrical or tapered or conical surface. Figure 1a shows the nomenclature and the identification forms for screw threads [1–4]. Threads are produced externally or internally (as shown in Fig. 1b) by cutting with a lathe-type tool: The process is called thread cutting or threading. When cut internally with a special threaded tool (tap), it is called tapping. External threads may also be cut with a die or by milling. In addition, traditional screw threads generation process is usually represented in two-dimensional sketch, but it may be difficult to understand the complex geometries and the manufacturing arrangement with the help of 2D models. These limitations can be partially overcome, and under-

Fig. 1 **a** Standard nomenclature and identification form for screw threads. **b** Diagram for screw thread cutting process



(*a*) Standard nomenclature and identification form for screw threads



(*b*) Diagram for screw thread cutting process

standing will be more meaningful if one uses 3D solid models instead. However, the development of the models using 3D models may not always ensure the clarity of the screw threads generation process unless one uses imitated animation (e.g., virtual reality (VR)) to represent the motion of the screw threads and cutter. Generation of screw threads can be achieved very efficiently with the assistance of the virtual manufacturing technique that is based on knowledge and expertise. It will enable the user to apply them more helpfully in real manufacturing situation.

As for the virtual environment (VE) technology, it has attracted both academics and industrialist to adopt VR in training and education. From the technological viewpoint, VR is usually a collection of advanced computer technology components and equipment. From the functional viewpoint, VR can be seen as a tool to provide real-time interactivity between computers and users. Through the dynamic display of computer graphics, users can interact with virtual objects and information in the VR environment. Meanwhile, VR is a tool that allows users to experience telepresence. Telepresence refers to "the extent to which one feels present in a computer-mediated environment, as well as in the immediate physical environment" [5]. According to these perspectives, we can define VR as a concept or tool consisting of computer technology-based artificial world filled with computer-generated images that respond to users' movements and inputs and which allows users to experience a mediated sense of presence. Virtual reality environments for training are proposed and fall into three categories [6]: networked text-based virtual environments, desktop virtual reality, and immersive virtual environments. Networked text-based virtual environments are commonly known as MUDs and MOOs [7] and support real-time interactive use among users in distributed and remote locations. Users basically interact with each other through typing words into their personal computers. Desktop virtual reality provides 3D multimedia simulations in which users can enter and explore. Users share the same virtual worlds through networking from remotely located personal computers [8]. Immersive virtual environments employ high-end laboratory equipment such as workstation computers, head-mounted display, and data gloves, and users are required to use laboratories and wear the equipment to interact with the virtual worlds. Although these environments provide some degree of immersion and thus may engage tutees attention and enhance learning, the cost—of hardware as well as software—limits their applications and popularity. A comparison of the three categories of virtual reality, in terms of their medium forms, playback platforms, development costs, etc., with respect to the VR system, is listed in Table 1. Through spatial, sensory, and audio simulation, a virtual environment can produce immersion effects. Therefore, it is also called "3D real-time interactive environment" [9–11]. It is suggested to construct virtual learning environments which can give learners close-to-reality perceptions and accumulate learning experiences through their interactions with the computer. Afterward, they will be able to apply their learned knowledge and competence in real-life conditions and complete their assigned tasks. A virtual reality system should be composed of three elements (the so-called 3Is) [12]: (1) immersion—users must be immersed into the virtual environment as if they were in the real-life environment; (2) interaction—the system must be able to detect users' actions and produce corresponding reactions in real time; (3) imagination—since virtual reality is a world built upon imagination, it can have more possibilities than merely simulating the real world. In summary, there are several characteristics listed: (1) the advanced visualization concept based on a virtual reality environment, (2) direct interaction with virtual objects in the holo-graphic visualizations, (3) multimedia integration (video, audio, text) in the interactive graphic simulation, (4) link to commercial CAD systems of the manufacturing domain [13].

In virtual manufacturing field, rapid prototyping, which is a new technology for design, is created [14] for visualization and verification. Graphical user interfaces virtual reality technologies, distillation, segregation, and auto interpretation are some of the important features of their work. A virtual prototyping is constructed using simulation of the planned production process using virtual manufacturing on a platform of MAYA, 3D Studio Max (3DS), and VRML [15]. Meanwhile, a research work is being carried out to simulate the gear manufacturing processes using AUTOCAD and 3DS as platforms [16, 17]. Software has been developed that helps the design engineers to understand the problems related to spur gear operation and its manufacturing process. An interactive design visualization system to facilitate an interactive product design at the conceptual design stage [18]. A modeling and simulation system is proposed to support a distributed machine design and control paradigm [19]. Meanwhile, the real-time simulation and validation has been extensively adopted in network-based VM environments. It is capable of generation information about the structure, status, and behavior of a manufacturing process and system [20].

From literature reviews, it is obvious that VR as a learning or training tool can make abstractions more concrete [21] and thereby help manufacturing companies internalize their learning [22–24]. By using head-mounted display and a 3D mouse, trainees could "touch" and "grasp" virtual objects in the learning center, determine the structural stability of various structures, visualize the internal forces on diagrams, and observe structural deformations. VR provides multiple or alternative representation of the real world. Participants and virtual objects are not constrained by physical realities or practicalities [25–28]. Meanwhile, VEs provide a means for users to navigate and investigate an efficient decision-making over entire process of manufacturing.

As mentioned above, the scope of virtual manufacturing is wide open for simulating assembly, machining, process planning, and so on. Computer simulation can be very effectively used for viewing along with aiding subsequent analysis of different complicated manufacturing processes. Hence, the purpose of this paper is to simplify the task of designing, study the screw threads generation process that can be understood by a learner (e.g., student and novice), and present a realistic view of it. This proposed system integrates the modules, users, and resources, and all the processes are developed by the OOP design, relational database management system, and the virtual reality technology. The overall structure of the paper is as follows: Section 2 presents the overall approach to the development of the web-based virtual manufacturing environment; Section 3 describes detailed approaches to the manufacturing tasks schedule model in VM environment over the

Table 1 Comparison of virtual reality systems

	Text-based	Desktop	Immersive
Medium forms	Test only	Dynamic 3D graphics, sound video	Dynamic 3D graphics, sound video
Playback platform	PC with network	PC with network	High-end computer with equipments, etc.
Playback platform cost	Low	Low to medium	High
Content development cost	Low	Medium	High
Immersive effects	Low	Low to medium	Medium
Mobility	Low	Low	Partial

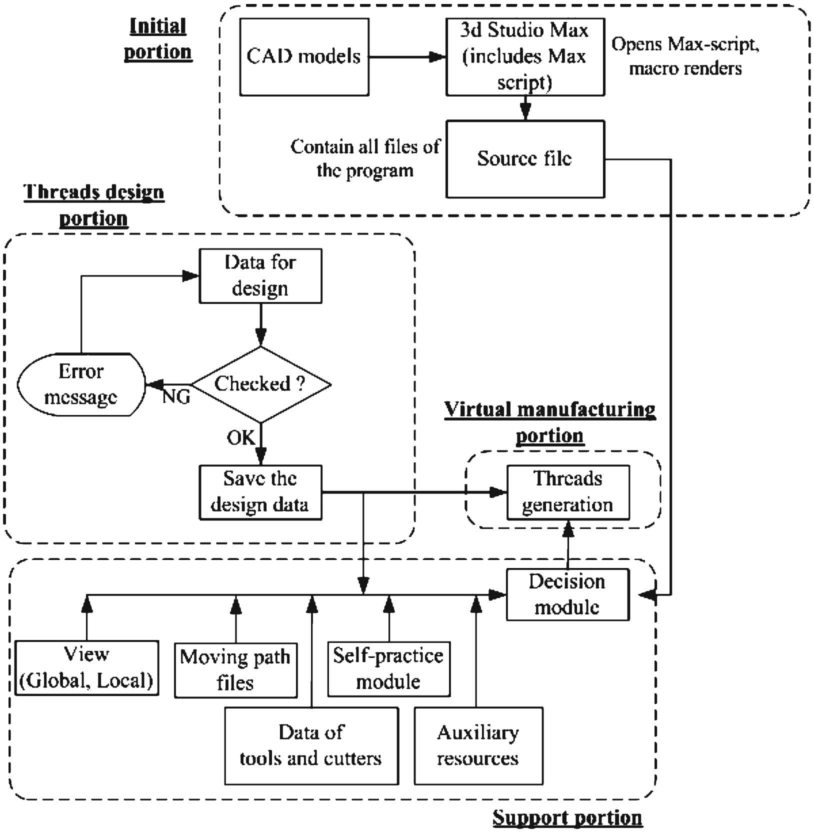
internet; Section 4 covers some examples and illustrations; concluding remarks are addressed in Section 5.

2 Development of a web-based virtual manufacturing environment

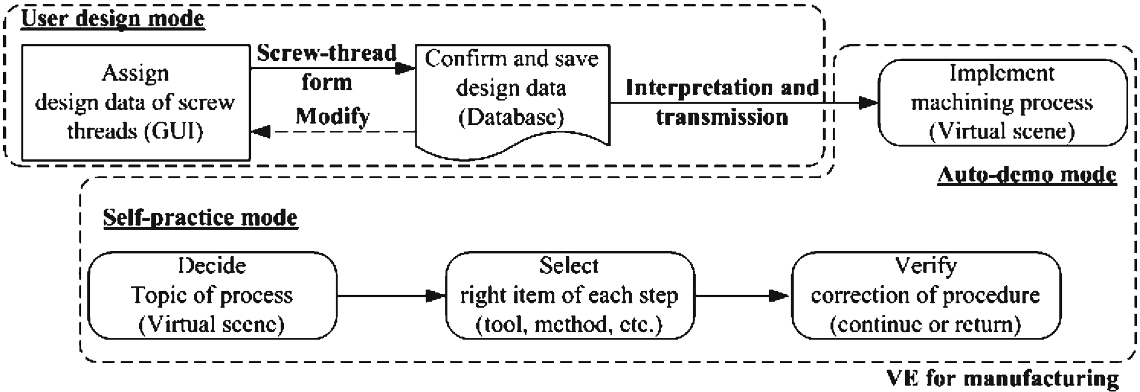
The entire structure of virtual manufacturing system is presented simplify in Fig. 2a. Execution sequence is shown in Fig. 2b. Both user design mode and self-practice mode allow users to understand the machining process of screw threads and facilitate hands-on operation, reducing the risk of damages of machine and operators injures. The key modules, initial, threads design, virtual manufacturing, and support four portions are included in this system. Explanations for these portions are detailed below.

2.1 Initial portion

This portion is created to offer the scene and essential objects. The former means the decoration of manufacturing workshop that includes, among others, working desks,



(*a*) The modular structure of virtual manufacturing system



(*b*) The executed procedure of virtual manufacturing system

Fig. 2 **a** The modular structure of virtual manufacturing system. **b** The executed procedure of virtual manufacturing system signboard, bulletin board; the latter includes equipment, tools, and the other auxiliary instruments, e.g., lathe, holder of straight chasers, vise. Meanwhile, the max script is basically an image processor that builds the visual effects (rendering, patch of material, and texture) in 3DS. Besides, it can be used for assignment of process sequences and repetitive tasks. As for the source files, these are made for the requirement of threads design portion, for example, assignment of initial conditions for manufacturing, standard template library of machining.

2.2 Threads design portion

This portion provides the parameters that are essential for the design and generation of the screw threads. The process of selecting the items and inputting the data is specifically finished through the html pages with the html tag. There are several key features that are based on the identification of threads have to be assigned through a series of dialog windows—type, unit, position, and the corresponding data (as shown in Fig. 3a). In addition, Fig. 3b illustrates that a user may specify the operational conditions that include the types of cutter. Other data relevant to machining process, such as feed rate, axis revolution, and dynamic load, will be automatically converted from user inputs. This module comprises the following steps:

1. Assignment of design parameters—The system will automatically retrieve and import the data from its database that correspond to the parameters of the thread input by the user. The data will be displayed to the screen for user's further confirmation (Fig. 1). The system also allows the user to input new parameters (use define mode), in which case the system will save the database with new parameters.
2. Confirmation —After designed parameters have been selected or entered (user define mode) by the user, the system will display screw-thread forms for user's confirmation.
3. Storage—When screw-thread data are confirmed, they will be stored in the database for the references in future machining process.

The system will automatically run the following procedure when the above steps have been finished: (a) open the execute file of VE, (b) declare variables and connect the database, (c) create the dynamic SQL query to research the key fields in the database, (d) save the multidimensional array in VE buffer, (e) close the connection and release buffer memory, (f) display the information of screw-thread form on the bulletin board in VE, (g) implement the process for cutting screw threads.

2.3 Virtual manufacturing portion

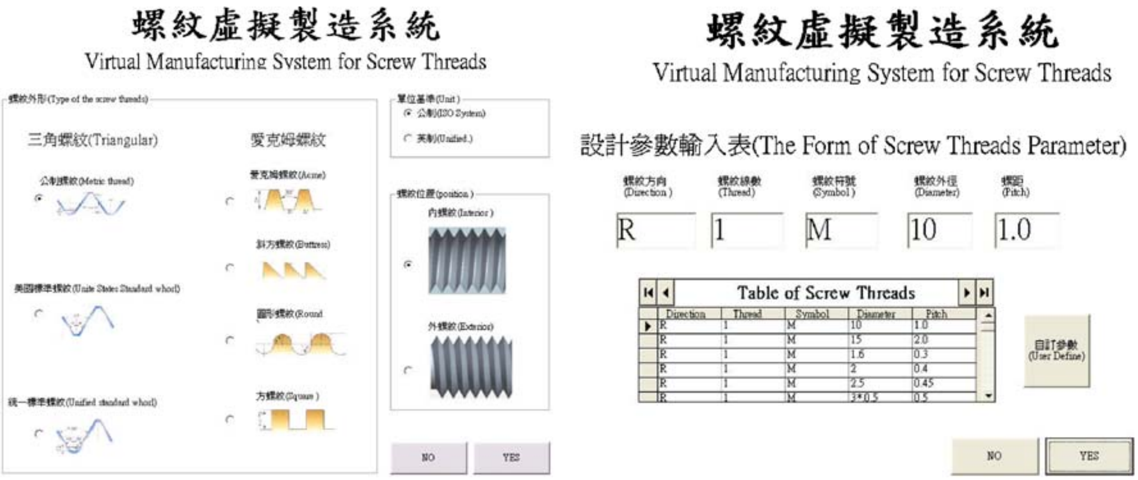
The physical models created in Section 2.1 will be imported to 3DS for the deployment of virtual manufacturing scene and the addition of texture. When completed, the scene and all components are imported into EON Studio and SDK development module to set up important nodes and parameters of the object attributes, light source, view angels, and motion in order to achieve the best visual effects. The brief construction process is shown in Fig. 4. As for the virtual manufacturing system of screw threads, there are user design mode and self-practice mode in the system.

2.3.1 User design mode

The cutter with all its cutting geometry depends on the form of screw threads; the workpiece is created following geometric conditions. These data have been provided by threads design portion. As for the screw threads generation, take an example of external threads with lathe cutting, the workpiece is clipped on the chuck. The tailstock supports the other end of the workpiece. It is equipped with a center that is fixed. The single-point cutting tool is positioned at square turret. Afterward, the cutter is given requisite motion and fed to generate profile threads. The process of machining screw threads is shown Fig. 5: (1) The decision module receives the data about the thread to be made, including screw threads form, data_tool, data_workpiece, and relevant parameters (axial revolution, feed rate, and cutting length, etc.), interprets them into SQL syntax, and finds the corresponding procedure (**field** no_topic); (2) the database puts the content_step in its designated place through VR operator and displays it in VM scene; (3) when VR operator finishes it task, finish_step message will be sent to decision module, then it will send a next_step request and finds the next process from the database; (4) steps (2)–(3) will be repeated until the machining comes to an end, at which time the decision module will send the "message_return" to Type library sender to terminate the user design mode.

2.3.2 Self-practice mode

Self-practice mode is entirely conducted in VE as shown in Fig. 6: (1) The user chooses the topic for the thread machining practice through VR interface. Decision module receives the id_practice of the user's choice and renders the topic as checked (distinguished by color t_click). Decision module then interprets it into SQL syntax and locates the corresponding **field** no_topic in the database, at which time system will lock other topics, allowing only one topic for each practice session; (2) the database puts relevant



螺紋虛擬製造系統

Virtual Manufacturing System for Screw Threads

螺紋外形(Type of the screw threads)

三角螺紋(Triangular)
- ● 公制螺紋(Metric thread)
- ○ 美國標準螺紋(Unite States Standard whorl)
- ○ 統一標準螺紋(Unified standard whorl)

愛克姆螺紋
- ○ 愛克姆螺紋(Acme)
- ○ 斜方螺紋(Buttress)
- ○ 圓形螺紋(Round
- ○ 方螺紋(Square)

單位基準(Unit)
- ● 公制(ISO System)
- ○ 英制(Unified)

螺紋位置(position)
- ● 內螺紋(Interior)
- ○ 外螺紋(Exterior)

螺紋虛擬製造系統

Virtual Manufacturing System for Screw Threads

設計參數輸入表(The Form of Screw Threads Parameter)

螺紋方向 (Direction)	螺紋線數 (Thread)	螺紋符號 (Symbol)	螺紋外徑 (Diameter)	螺距 (Pitch)
R	1	M	10	1.0

Table of Screw Threads

Direction	Thread	Symbol	Diameter	Pitch
R	1	M	10	1.0
R	1	M	15	2.0
R	1	M	1.6	0.3
R	1	M	2	0.4
R	1	M	2.5	0.45
R	1	M	3*0.5	0.5

(*a*) Dialogue window

螺紋虛擬製造系統

Virtual Manufacturing System for Screw Threads

加工刀具(Cutter)及加工設備(Cutting tools)

加工刀具(Cutter)
- ○ 內牙刀(Internal single-point cutting tool)
- ○ 螺絲攻(Straight chasers)

加工設備(Cutting tools)
- ○ 車床(Lathe)
- ○ 螺絲攻把手(Holder of straight chasers)

(*b*) Parameters assignment

Fig. 3 **a** Dialog window. **b** Parameters assignment

content_step and item_step to designated place through VR operator; (3) the user selects the most suitable answer (id_item_step) according to the question of the step. Then, the answer is sent for decision module to check, and the item rendered is selected (distinguished using colors, s_click); (4) if the item is correct, decision module will send a next_step request, finding the next step from the database; (5) steps (2)–(4) will be repeated until the machining process comes to an end; (6) if the item is incorrect, decision module will send a id_item_step request and locates the corresponding error message (**field** alarm) and prompts contents (text_result), which will be sent and displayed in the **TextBox** of VR operator.

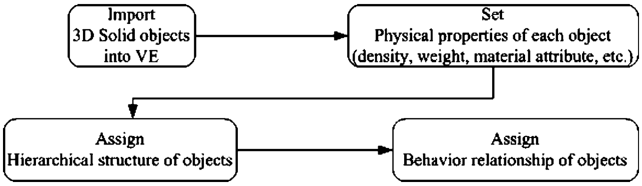
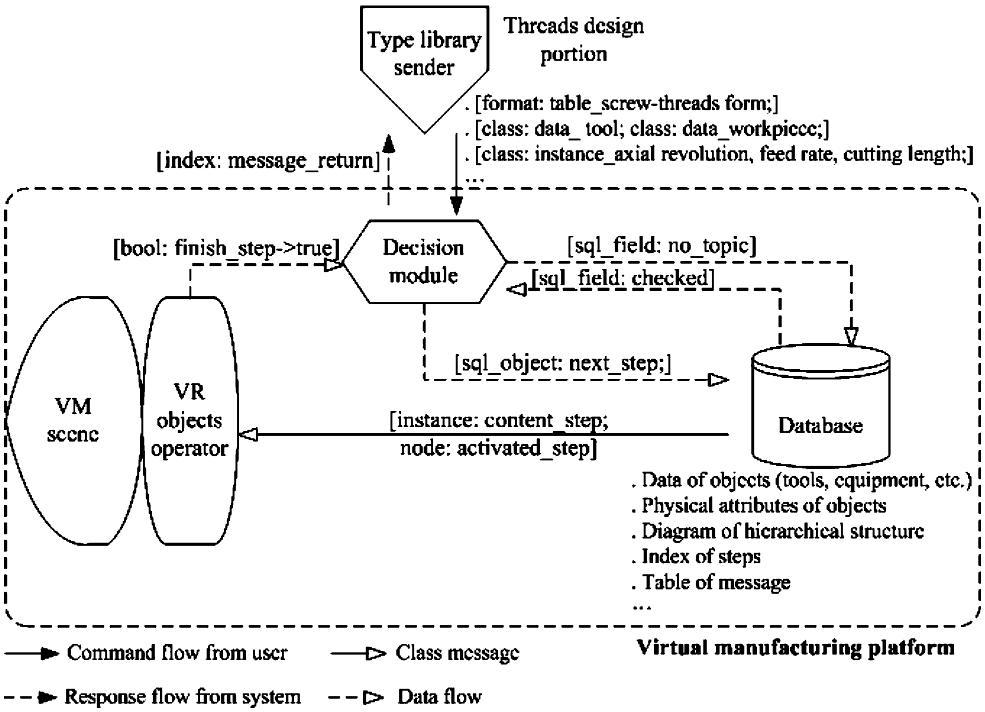


Fig. 4 Flowchart for building a virtual manufacturing environment

2.4 Support portion

There are several features developed in this portion: They are camera viewing, animation path, self-practice, and auxiliary sources. The camera viewing module provides the facility to place the camera at different coordinate positions (global and local two types) and thus show different camera views of the machining process. Still pictures of the partially cut workpiece along with that of the cutter at every step of cutting is recorded and enable the

Fig. 5 Diagram for process of machining screw threads—user design mode



user to feel the reality in a virtual environment. The animation path module records the moving trace and its frames that are displayed in proper sequence at successive interval and simulates the dynamic behavior of different components. The cutter and the workpiece occupy different positions in each of the frames depending on the kinematic relationship of the machining process (e.g., feed rate, axial revolution, machining stroke, etc.).

2.5 Architecture of rational database management system

A three-tier architecture on the internet is used to implement the tasks schedule model in this paper. The architecture consists of a web server, a DB server, and clients. A rational database management system (RDBMS), MySQL, is used to build the remote DB server. The MySQL provides standards-based drivers for JDBC and open database connectivity, enabling the developer to build DB application in their language. An RDBMS resides on a tier of the multi-tier architecture, which is farthest removed from the end users. JDBC techniques aid clients to connect to a remote DB server through internet. The database structure shown in Fig. 7 consists of several tables and relationships conceptually. The tables are designed to store hierarchical object information. In addition, a DB importer is implemented on the DB server to efficiently store the data of VR objects into the DB that requires organized and structured information. The DB importer has the following functions: (1) search the name of screw threads (table **Screw_Threads**) and its corresponding tools for machining

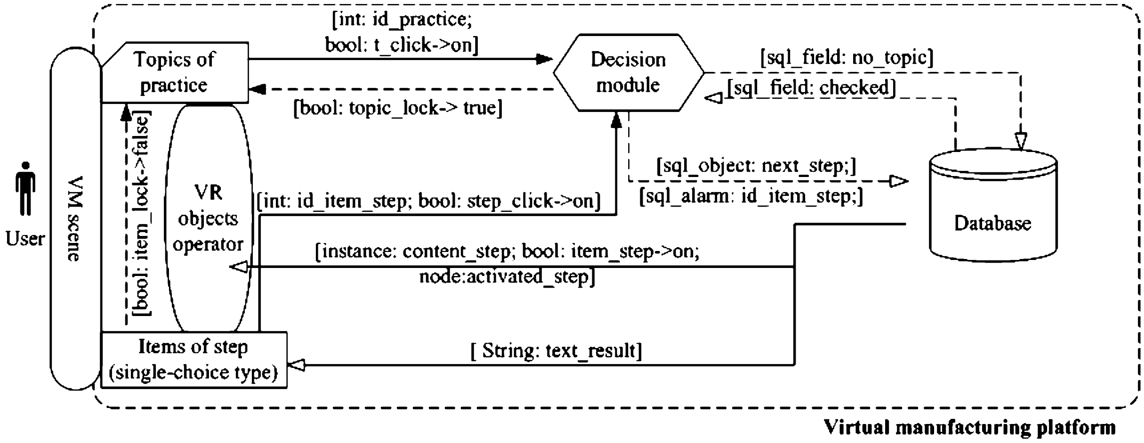


Fig. 6 Diagram for process of machining screw threads—self-practice mode

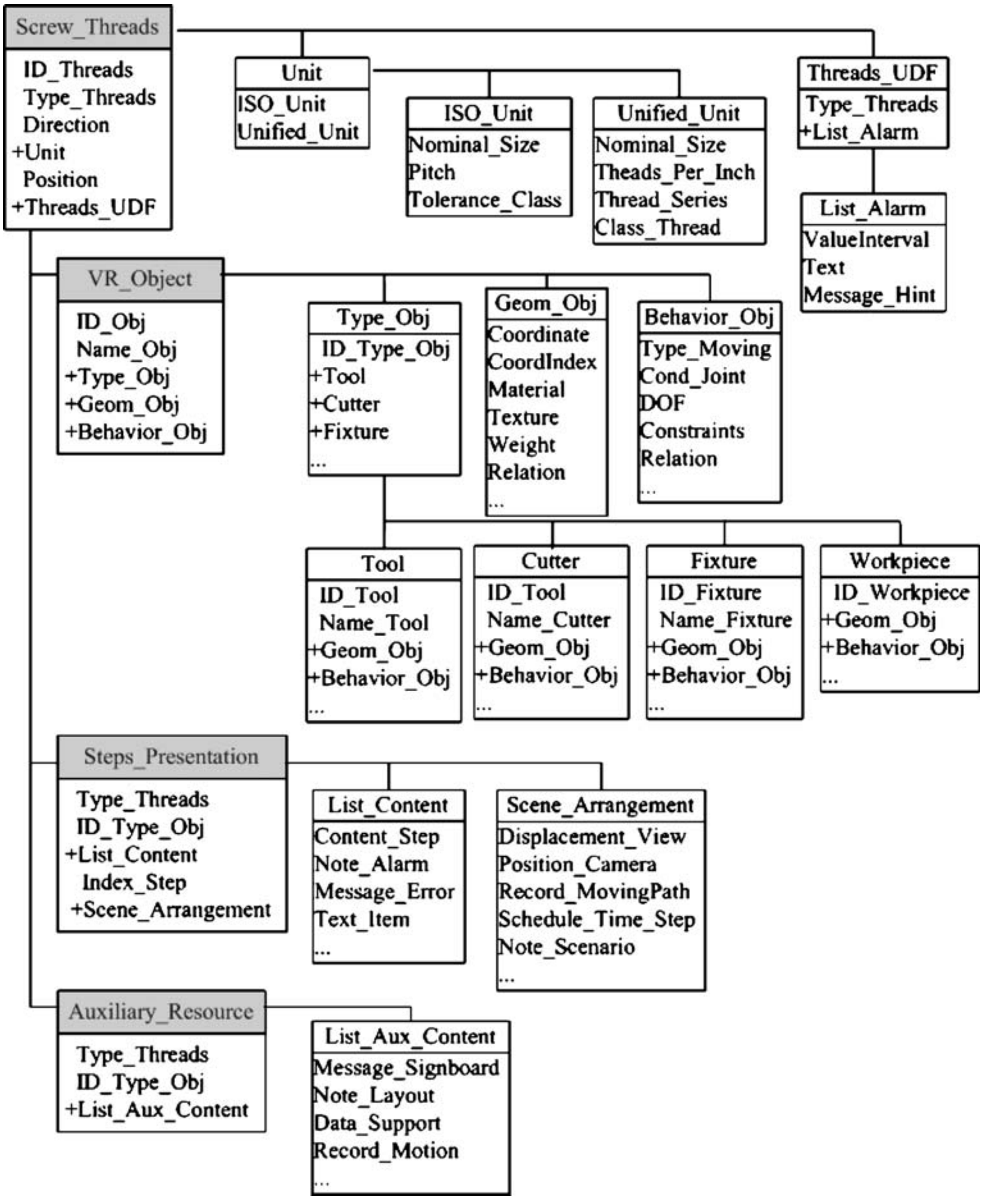


Fig. 7 Conceptual diagram of relational database model for machining screw threads

(table **VR_Object**); (2) open the file and transform the coordinates of VR object; (3) build a dynamic SQL query to insert the design data into the database; (4) connect and transact to the database.

2.6 Communication of application interface

On the server side, the programs are defined as the handler with JDBC and socket handler and serve to, respectively, communicate with the remote DB server and clients over the internet. Meanwhile, these programs are implemented on the http web server. The server has the following characteristics: (1) receive the connection from the browser; (2) run application programs for common server technologies such as hypertext pre-processor, java server page; (3) transmit data back to client and record a log of user activity.

On the client, an html file transmitted from the server runs on a web browser. The html file embeds several java applets and browsers serving as Rpaptor (a free 3DS plug-in). A java applet communicates with the web server by means of the java socket-based communication and interacts with the VR objects in the browsers. A client application consists of several java applets based on their functionality. Furthermore, applets include server selector, message windows, VR object handler, and the other applets developed for tasks schedule model. These applets are formed as objects or components so that a web-based client application can be easily implemented. Besides, it is possible for an applet method in a standard web browser to access the VR object of an embedded Raptor as illustrated in Fig. 8. There are two key functions provided by EAI: (1) read external programs and change the VR object and (2) register the functions of external programs as

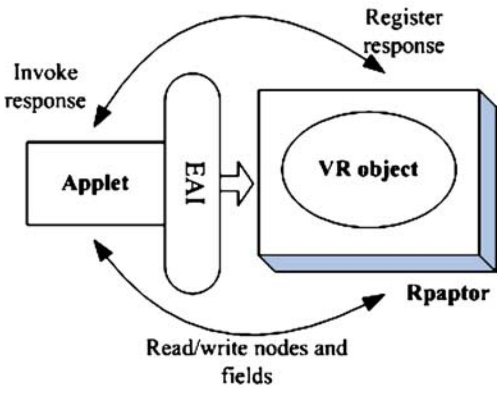


Fig. 8 Communication between a java applet and Rpaptor in client application

responses. Whenever this event (i.e., response) is generated in the VR environment, the browser invokes the associated return function and passes the current value of the event as a parameter.

The communication between a server and clients in this paper, the java socket-based method is used. The communication is programming language independent so that a socket program written in java also communicates to a program written in non-java socket class. Meanwhile, the format of command protocol is composed of **ID_Threads**, **Name of DB**, **Amount of argument**, and **List of arguments**. **ID_Threads** specifies a form to be invoked and executed in the server; by **Name of DB**, a specific RDB, including various tables and relationships, is subsequently selected in the designated DB server. **Amount of argument** and **List of arguments** are used to form a dynamic SQL query to get intended result in transacting with the RDB.

3 Task-oriented decision approach to the virtual machining process

Tasks decision of product aims to meet demands of feasibility customization, cost effective, right delivery and service and environmental satisfaction in a product life cycle [29]. An automated and computerized assembly planning is integrated with CAD systems [30]. The importance of a number of assembly tools applied and tool changes in achieving optimal assembly planning is discussed [31]. Disassembly planning aims to generate disassembly sequences with subsequent disassembly actions, which has been addressed by several researchers [32]. As for the methodology for tools selection in assembly planning, a system tools selection is developed based on general guideline rules [32]. A systematic method for identifying and ranking tool alternatives is based upon production cost and time [33]. An incorporated tool selection system is developed to sequentially determine tool types by means of object-oriented knowledge and DB search [34]. Furthermore, machine selection is also a critical activity of job scheduling that is defined as the allocation of resources over time to perform a collection of tasks [35, 36]. The available machines and possible operations could be involved in machine selection [37].

A tasks schedule model is generated by decision module in virtual manufacturing environment as illustrated in Fig. 9. It can be created based on the available resources, which change along with a screw threads machining process. This section explains detailed approaches proposed to tasks schedule model in VM environment.

3.1 Module architecture

The three-phase modeling framework is applied in this paper [38]. The modeling framework is based on the object-oriented modeling paradigm [39] and consists of three sub-modules—static module (virtual component), flow module (part flow), and administered module (control the part flow). All virtual components are built in Section 2, e.g., tools, cutters and facilities. Part flow is visualized by VR operator that is generated by java applets and EAI. As for the administered module (control the part flow), it is implemented following the results of the mathematical formalism.

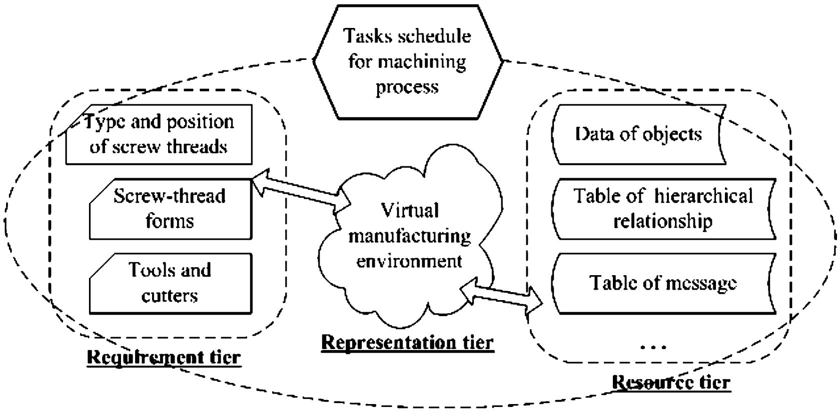


Fig. 9 Tasks schedule based on virtual manufacturing environment

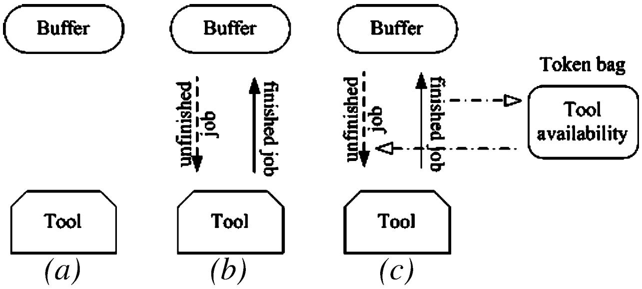


Fig. 10 Three-phase modeling procedure

Figure 10 (a) shows the static module, which is constructed by positioning two device symbols, a buffer and a tool. The flow module, illustrated in Fig. 10 (b), describes the part flow in the system by drawing arrows between device symbols. The solid line represents the flow of a finished part from the tool to the buffer, and the dotted line represents the flow an unfinished part from the buffer to the tool. Figure 10 (c) shows the administered module, which defines the firing conditions of each flow according to the system state. For instance, the token bag, shown in Fig. 10 (c), represents the availability of the tool to control the flow of unfinished part from the buffer to the tool.

The static module consists of manufacturing tools having their positions in the layout. The flow module is supposed to describe the flow (or transfer) of parts and auxiliary resources (pallets and holder, etc.) between tools. A transfer is modeled as an object, called transfer operator, which includes a set of tool-level orders enabling the transfer. The administered module states the control logic of a manufacturing process, which makes decisions on friable transfers based on the decision parameters, which can express meaningful system states. Two objects are employed for the administered module—flow controller and state manager. The former is used for making decisions on friable transfers based on the decision parameters. The latter is applied for maintaining decision parameters based on the mapping relation between decision parameters and the states of virtual components in the system. The overall structure of the proposed decision module for creating the tasks schedule model is shown in Fig. 11. The virtual component provide interface with two sub-modules, state manager and transfer operator, in the decision module. While the relation between virtual component and state manager is a one-way communication, the communication between the virtual component and transfer operator should be two-way. The transfer operator sends component-level orders to the virtual component, and it also receives a feedback from the virtual component, e.g., the return of order (done or failure) of the command execution. The state manager maintains decision parameters representing the state of tasks schedule, and the flow controller chooses friable transfers according to the decision parameters.

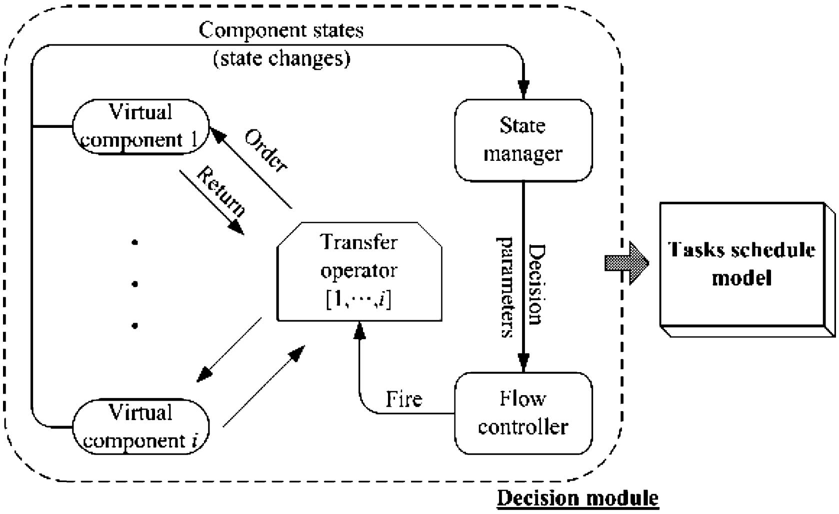
In this paper, the discrete event simulation formalism [40] is employed to support the specification of discrete event models in a hierarchical, modular manner. Meanwhile, the semantics of the formalism are compatible with object-oriented specifications for simulation models. Virtual manufacturing environment is a simulated platform, which realizes the formalism for modeling and associated abstract simulator concepts for simulation, all in object-oriented language. Within the formalism, there are two parts have to be specified: unit and pair. Formally, a unit ***UN*** is specified by a 7-tuple:

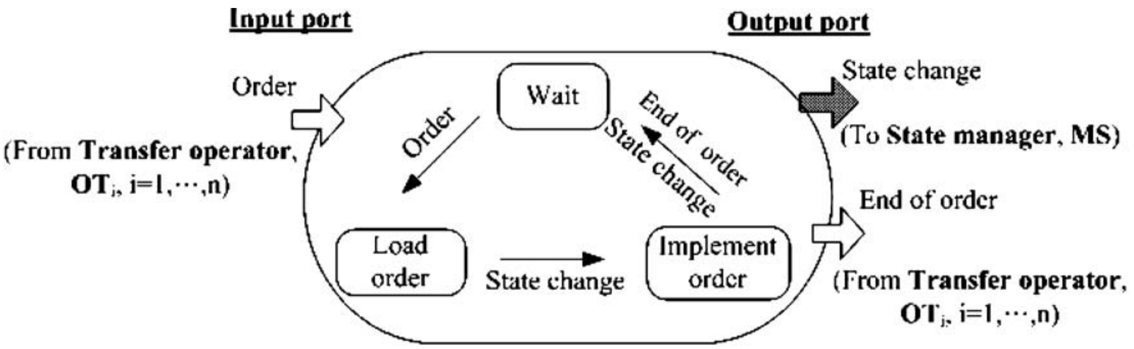
$$UN = \{E_{\text{in}}, S_{\text{ss}}, E_{\text{ex}}, F_{\text{t\_in}}, F_{\text{t\_ex}}, F_{\text{out}}, F_{\text{ad\_t}}\}$$

In this form:

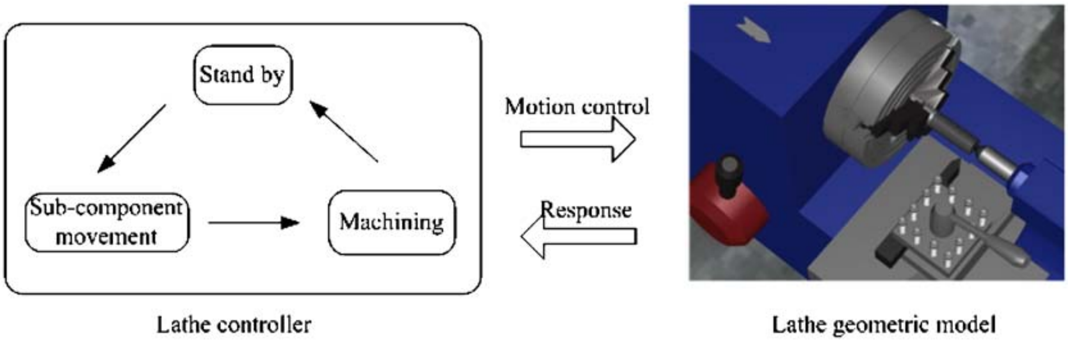
E_{in}: input events set
S_{ss}: sequential states set

Fig. 11 Conceptual diagram of decision module

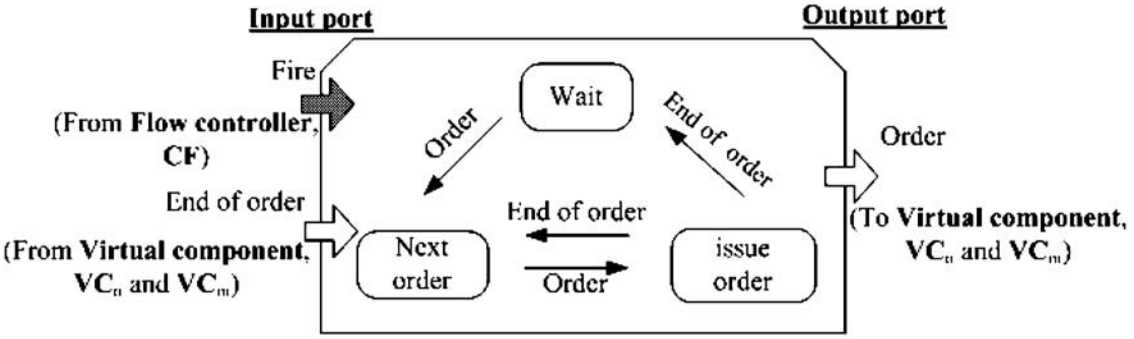




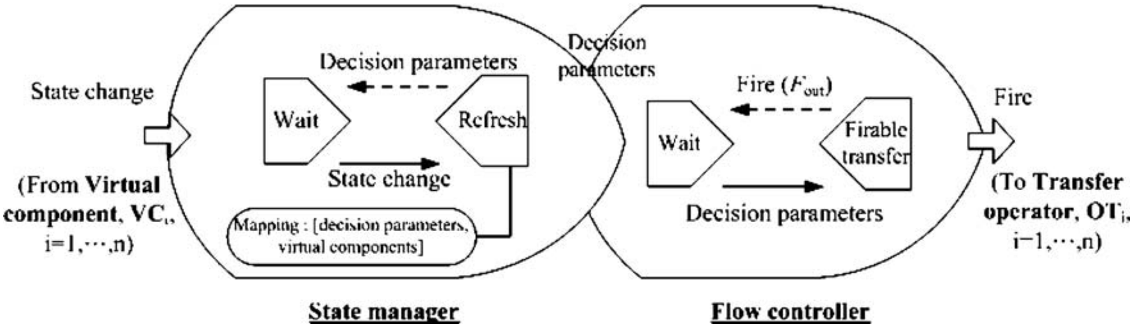
(*a*) State transition diagram of virtual component



(*b*) The controller of virtual component



(*c*) State transition diagram of transfer operator



(*d*) State transition diagram of state manager and flow controller

Fig. 12 **a** State transition diagram of virtual component. **b** The controller of virtual component. **c** State transition diagram of transfer operator. **d** State transition diagram of state manager and flow controller

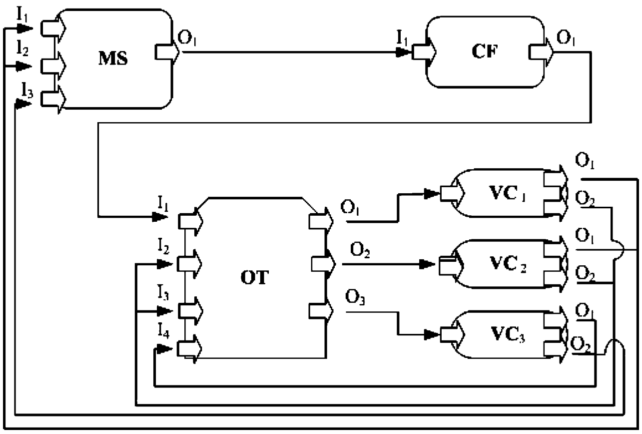


Fig. 13 Diagram of tasks schedule model

E_{ex}: output events set
F_{t_in}: $S_{ss} \to S_{ss}$: internal transition function
F_{t_ex}:$T_s \times E_{in} \to S_{ss}$: external transition function
$T_s=\{(s,ti)|s\in S_{ss},\ 0\leq ti\leq F_{ad\_t}\ (s)\}$: total state of ***UN***
F_{out}: $S_{ss} \to E_{ex}$: output function
F_{ad_t}: $S_{ss} \to$Real: time advance function

The four elements in the above form, F_{t_in}, F_{t_ex}, F_{out}, F_{ad_t}, are called the characteristic functions of a unit model. Figure 12a illustrates the state transition diagram of a virtual component have several related transfers, and the corresponding unit model can be depicted as follows.

$$UN = \{E_{in}, S_{ss}, E_{ex}, F_{t\_in}, F_{t\_ex}, F_{out}, F_{ad\_t}\}$$

E_{in}=[$\mathbf{OT}_1$,...$\mathbf{OT}_n$]
S_{ss}=[Wait, Load order, Order execution]

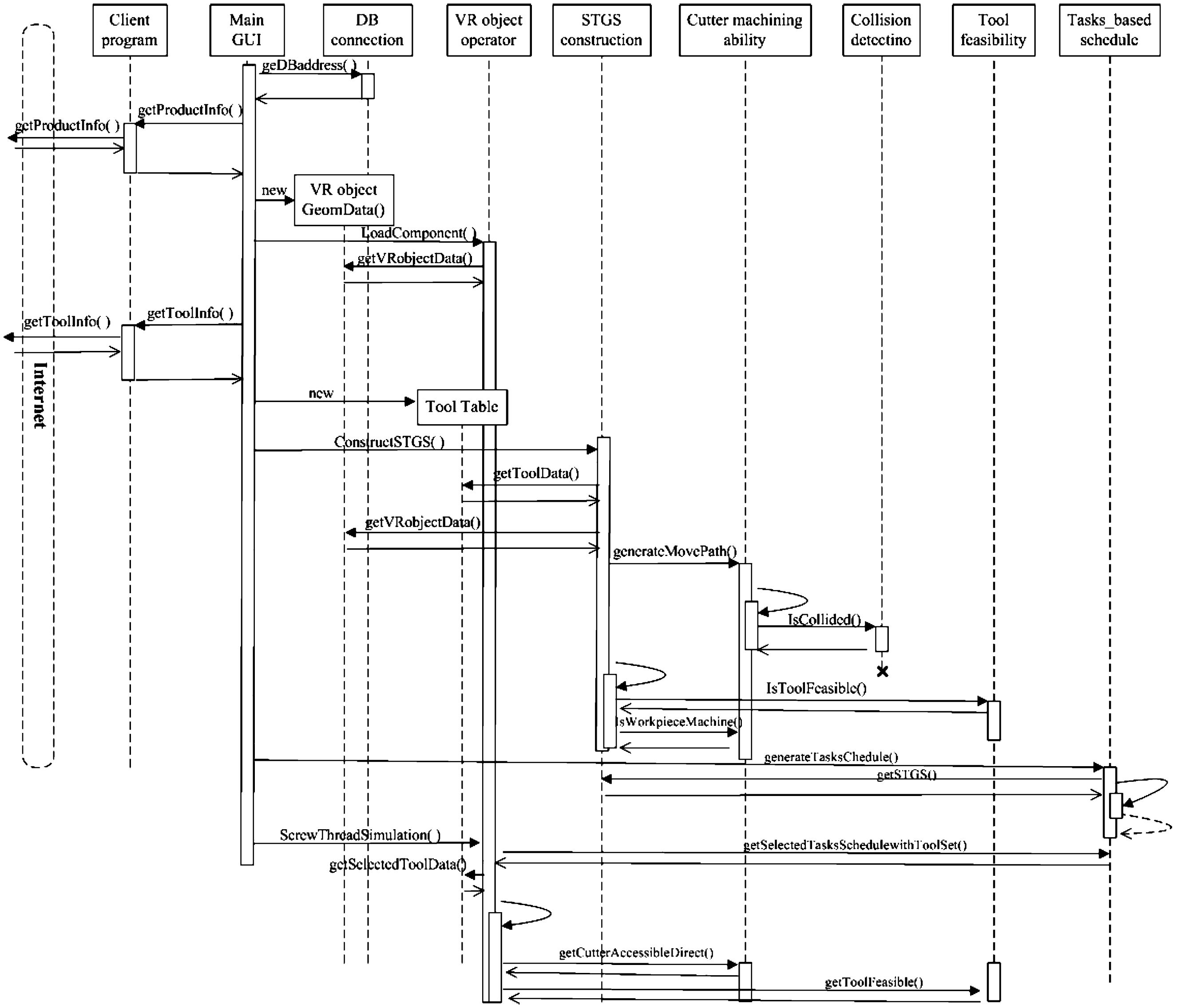


Fig. 14 Sequence diagram of creation of tasks schedule model

E_{ex}=[**MS**, **OT**$_1$,…**OT**$_n$]
F_{t_in}(Load order)=Order execution
F_{t_in}(Order execution)=Wait
F_{t_ex}(Wait, **OT**$_1$,…,or **OT**$_n$)=Load order
F_{out}(Load order)=**MS**; F_{out}(Order execution)=**OT**$_1$,…, or **OT**$_n$; F_{out}(Order execution)=**MS**
F_{ad_t}: $S_{ss}\rightarrow$Real: time advance function

As for the pair model ***PA*** is defined as:

$$PA = \left\{E_{\text{in}}, E_{\text{ex}}, UN, R_{\text{in\_ex}}, R_{\text{out\_ex}}, R_{\text{p\_in}}, S_{\text{pick}}\right\}$$

E_{in}: input events set
E_{ex}: output events set
UN: set of all models in discrete event system
$R_{in\_ex}\subseteq \boldsymbol{PA}.\text{in}\times \boldsymbol{UN}.\text{in}$: external input pair relation
$R_{out\_ex}\subseteq \boldsymbol{PA}.\text{out}\times \boldsymbol{UN}.\text{out}$: internal output pair relation
$R_{p\_in}\subseteq \boldsymbol{UN}.\text{out}\times \boldsymbol{UN}.\text{in}$: internal pair relation
S_{pick} :$2^{\boldsymbol{UN}}-\Phi\rightarrow \boldsymbol{UN}$: tie-breaking selector

Where the .in and .out mean the input ports set and the output ports set of respective discrete event model. The four modules—virtual component, transfer operator, state manager, and flow controller—will be represented as unit models, and the whole tasks schedule model will be included in the unit models and pair relations between them.

3.2 Virtual component and transfer operator

The virtual component encloses the key data, which represents the inherent properties of component, such as geometric shape, kinematics, and execution of component-level orders, besides the creation of CAD objects in Section 2.1. In the physical aspects, the virtual component have a geometric model with kinematics, such as moving joints (translation or rotation) and their attributes, including speed, feasible moving range, and so on. To control the geometric model, a component controller models the logical aspects of a component. The component controller should be able to implement component-level orders by operating the geometric model. An example of the virtual component is shown in Fig. 12b, the component controller of lathe should be able to interpret motion order of sub-components (Cartesian coordinate values), report state change, and control the geometric model.

A transfer operator needs to communicate with the flow controller and two virtual components (at least). In Fig. 12c, a transfer operator has an input port from the flow controller through which a firing signal activates the transfer. The transfer operator finds component-level orders, which will be delivered to corresponding virtual components through two output ports. When the virtual components complete the orders, they inform the transfer operator of the end of orders through two input ports. Afterward, the transfer operator finds the next order to be

螺紋虛擬製造系統
Virtual Manufacturing System for Screw Threads

Aims of System

Design mode：
This mode guides the novice or engineers through the several design process enabling the creation of tasks schedule model, its contents include the selection of tools and cutter, decision of methods and procedure for machining, and so on. Through the use of mode, a machining process of screw threads step by step in virtual manufacturing environment can also be implemented automatically.

Practice mode：
This mode evaluates the users' learning result in screw threads generation. There are several topics are developed following the various types of screw threads and theirs related data. Through the single-choice type per step in each topic, user is able to understand the whole machining process besides self-evaluation.

(*a*) Main webpage for screw threads generation

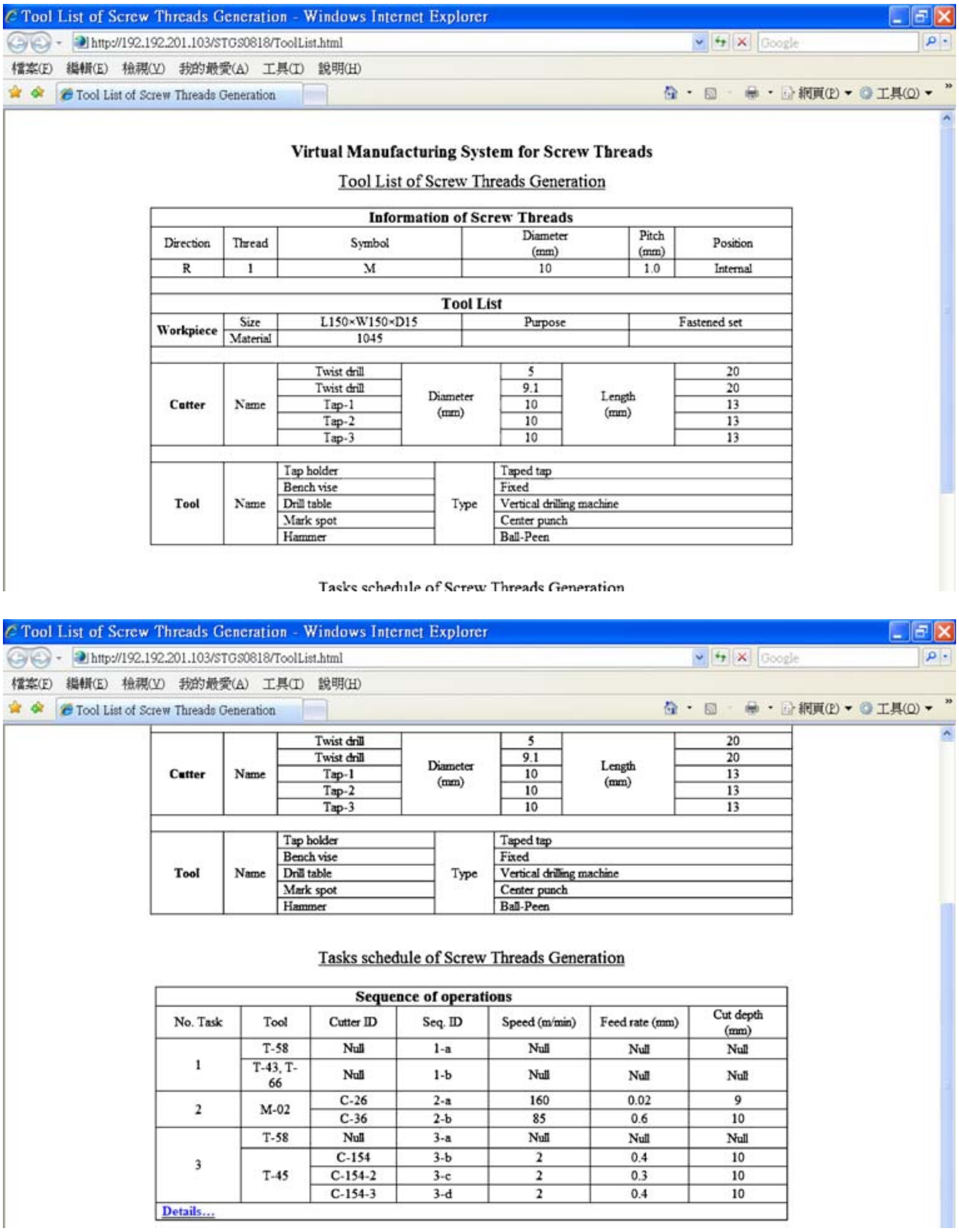
Fig. 15 **a** Main webpage for screw threads generation. **b** Information for screw threads generation. **c** Details of each task for screw threads generation. **d** Browsing mode for generating the screw threads executed. The above procedure is repeated until the transfer is completed. As for the unit model corresponding to the transfer operator, it can be depicted as follows:

$E_{\mathrm{in}}=[\mathbf{CF}, \mathbf{VC}_n, \mathbf{VC}_m]$
$S_{\mathrm{ss}}=[\text{Wait, Next order, Issue order}]$
$E_{\mathrm{ex}}=[\mathbf{VC}_n, \mathbf{VC}_m]$
$F_{\mathrm{t\_in}}(\text{Next order})=\text{Issue order}$
$F_{\mathrm{t\_in}}(\text{Issue order})=\text{Next order or Wait}$
$F_{\mathrm{t\_ex}}(\text{Wait}, \mathbf{CF})=\text{Next order}$
$F_{\mathrm{out}}\ (\text{Issue order})=\mathbf{VC}_n \text{ or } \mathbf{VC}_m$
$F_{\mathrm{ad\_t}}: S_{\mathrm{ss}}\rightarrow\text{Real: time advance function}$

3.3 State manager and flow controller

The state manager maintains decision parameters representing the screw threads machining state, and the flow controller chooses friable transfer according to the decision parameters. Figure 12d shows the state transition diagram. Whenever there is a state change of a virtual component, it should report to the state manager through the corresponding input port, then the state manager has to reflect the change the decision parameters and the state of virtual component. Meanwhile, any change in the decision parameters, the state manager should report to the flow

Fig. 15 continued.



Virtual Manufacturing System for Screw Threads

Tool List of Screw Threads Generation

Information of Screw Threads					
Direction	Thread	Symbol	Diameter (mm)	Pitch (mm)	Position
R	1	M	10	1.0	Internal

Tool List						
Workpiece	Size	L150×W150×D15	Purpose		Fastened set	
	Material	1045				
Cutter	Name	Twist drill	Diameter (mm)	5	Length (mm)	20
		Twist drill		9.1		20
		Tap-1		10		13
		Tap-2		10		13
		Tap-3		10		13
Tool	Name	Tap holder	Type	Taped tap		
		Bench vise		Fixed		
		Drill table		Vertical drilling machine		
		Mark spot		Center punch		
		Hammer		Ball-Peen		

Tasks schedule of Screw Threads Generation

Sequence of operations						
No. Task	Tool	Cutter ID	Seq. ID	Speed (m/min)	Feed rate (mm)	Cut depth (mm)
1	T-58	Null	1-a	Null	Null	Null
	T-43, T-66	Null	1-b	Null	Null	Null
2	M-02	C-26	2-a	160	0.02	9
		C-36	2-b	85	0.6	10
3	T-58	Null	3-a	Null	Null	Null
	T-45	C-154	3-b	2	0.4	10
		C-154-2	3-c	2	0.3	10
		C-154-3	3-d	2	0.4	10
Details...						

(*b*) Information for screw threads generation controller through the output port. As for the unit model corresponding to both, it can be described as follows:

State manager:

$E_{\mathrm{in}}=[\mathbf{VC}_1,\ldots,\mathbf{VC}_n]$, where n is the number of virtual components
$S_{\mathrm{ss}}=[\mathrm{Wait},\ \mathrm{Refresh}]$
$E_{\mathrm{ex}}=[\mathbf{CF}]$
$F_{\mathrm{t\_in}}(\mathrm{Refresh})=\mathrm{Wait}$
$F_{\mathrm{t\_ex}}(\mathrm{Wait},\ \mathbf{VC}_1\ \mathrm{or},\ldots\ \mathrm{or}\ \mathbf{VC}_n)=\mathrm{Refresh}$
$F_{\mathrm{out}}\ (\mathrm{Refresh})=\mathbf{CF}$
$F_{\mathrm{ad\_t}}$: $S_{\mathrm{ss}}\rightarrow\mathrm{Real}$: time advance function

Flow controller:

$E_{\mathrm{in}}=[\mathbf{MS}]$
$S_{\mathrm{ss}}=[\mathrm{Wait},\ \mathrm{Firable\ transfer}]$
$E_{\mathrm{ex}}=[\mathbf{OT}_1,\ \ldots,\mathbf{OT}_n]$, where n is the number of transfer operators
$F_{\mathrm{t\_in}}(\mathrm{Firable\ transfer})=\mathrm{Wait}$
$F_{\mathrm{t\_ex}}(\mathrm{Wait},\ \mathbf{MS})=\mathrm{Firable\ transfer}$
$F_{\mathrm{out}}\ (\mathrm{Firable\ transfer})=\mathbf{OT}_1,\ \ldots\ \mathrm{or}\ \mathbf{OT}_n$
$F_{\mathrm{ad\_t}}$: $S_{\mathrm{ss}}\rightarrow\mathrm{Real}$: time advance function

Whenever the stage manager reports a state change of the system, the flow controller makes a decision on friable transfers based on the flow control logic. If there is a friable transfer, the flow controller gives an order to fire the transfer through the corresponding output port.

Figure 13 illustrates a simplified example of a tasks schedule model for machining the internal screw threads by manual. There are two virtual components—holder of straight chasers ($\mathbf{VC}_1$), straight chasers ($\mathbf{VC}_2$), and vise ($\mathbf{VC}_3$), one transfer operator. The pair model corresponding to the tasks schedule model is described as follows:

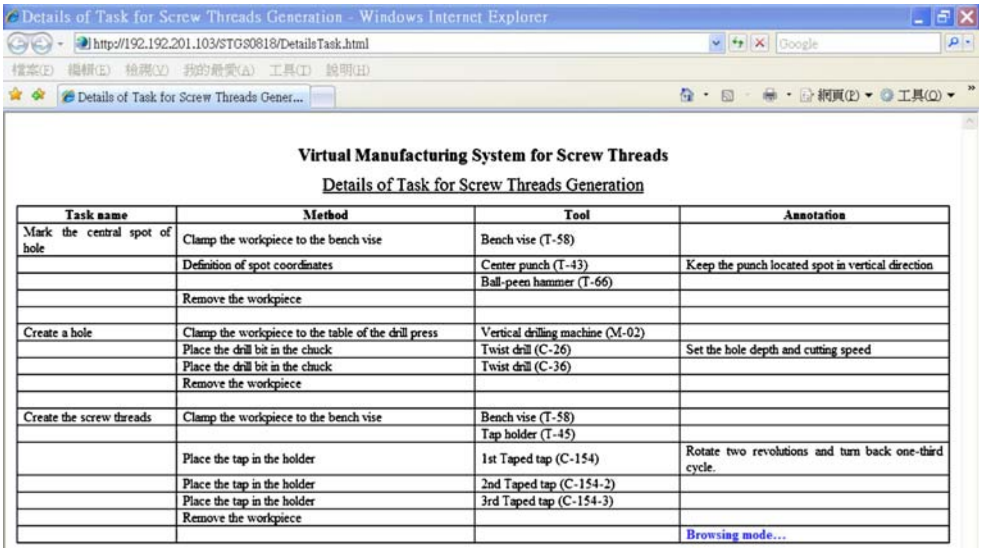
Tasks schedule model **TSM**: $\boldsymbol{PA}=\{E_{\mathrm{in}},\ E_{\mathrm{ex}},\ \boldsymbol{UN},\ R_{\mathrm{in\_ex}},\ R_{\mathrm{out\_ex}},\ R_{\mathrm{p\_in}},\ S_{\mathrm{pick}}\}$
$\boldsymbol{UN}=[\mathbf{VC}_1,\ \mathbf{VC}_2,\ \mathbf{VC}_3,\ \mathbf{OT},\ \mathbf{MS},\ \mathbf{CF}]$
$R_{\mathrm{p\_in}}=[(\mathbf{MS}.O_1\times\mathbf{CF}.I_1),\ (\mathbf{CF}.O_1\times\mathbf{OT}.I_1),\ (\mathbf{OT}.O_1\times\mathbf{VC}_1.I_1),\ (\mathbf{OT}.O_2\times\mathbf{VC}_2.I_1),\ (\mathbf{VC}_1.O_1\times\mathbf{MS}.I_1),\ (\mathbf{VC}_1.O_2\times\mathbf{OT}.I_2),\ (\mathbf{VC}_2.O_1\times\mathbf{MS}.I_2),\ (\mathbf{VC}_2.O_2\times\mathbf{OT}.I_3),\ (\mathbf{OT}.O_3\times\mathbf{VC}_3.I_1),\ (\mathbf{VC}_3.O_1\times\mathbf{MS}.I_3),\ (\mathbf{VC}_3.O_2\times\mathbf{OT}.I_4)]$
$S_{\mathrm{pick}}:2^{\boldsymbol{UN}}-\Phi\rightarrow\boldsymbol{UN}$: tie-breaking selector

In summary, a decision module is composed of four sub-modules. While the four sub-modules are implemented as unit models, the whole tasks schedule model is executed as a pair model, including those unit models and pair relations among them. The main objective of mathematical formalism is the decision of the whole manufacturing process. The essential data and references are retrieved automatically from the requirement tier and resource tier (as shown in Fig. 9). Then, the result is transformed into the task schedule model and is shown in format of html pages (i.e., tasks schedule table). Finally, the whole manufacturing process is visualized in the representation tier (as also illustrated in Fig. 9).

4 Implementation aspects of the system

In a web-based virtual manufacturing environment, the developed system is accessed by users on the client side. In planning tasks schedule model, a DB server, which stores

Fig. 15 continued.



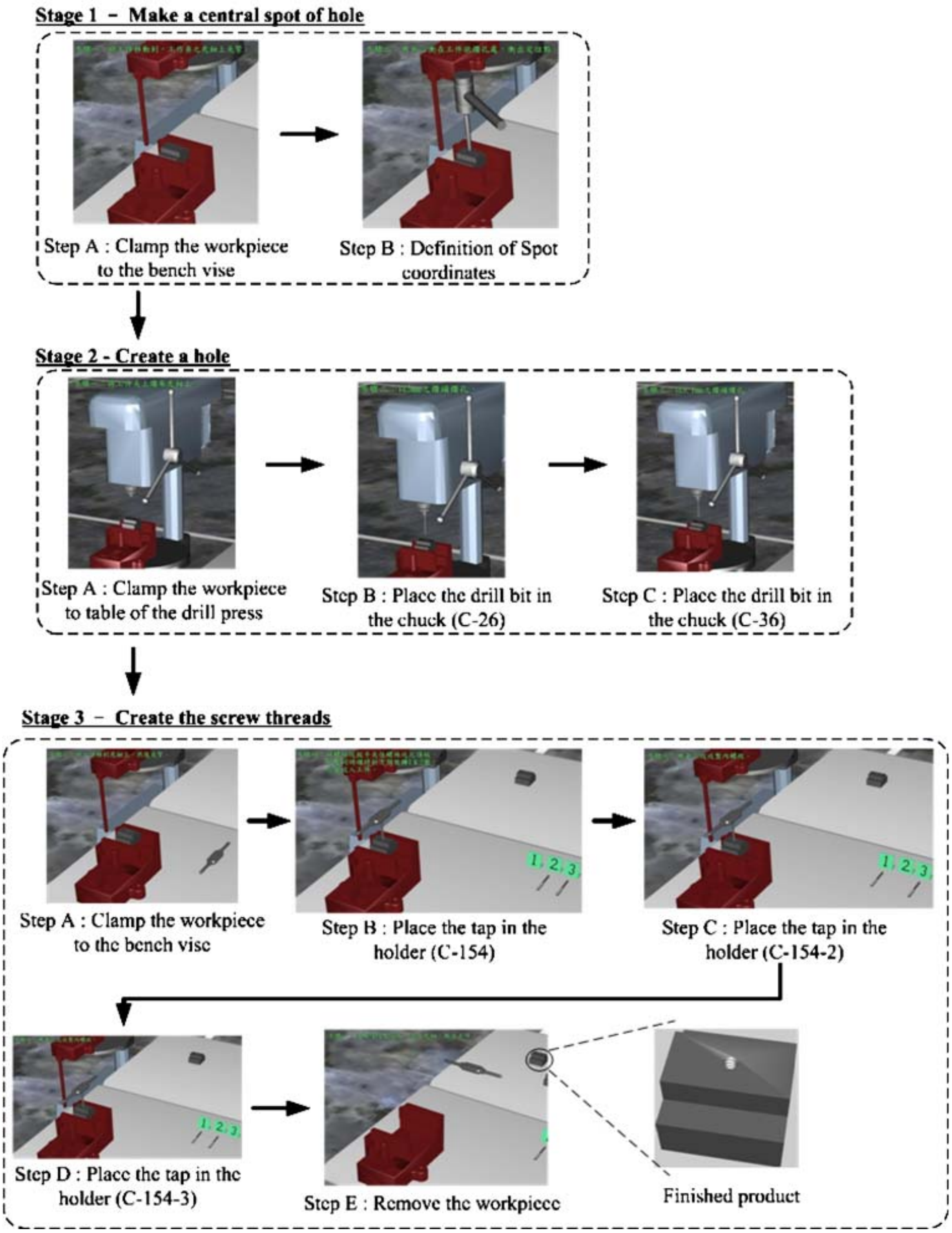
Virtual Manufacturing System for Screw Threads

Details of Task for Screw Threads Generation

Task name	Method	Tool	Annotation
Mark the central spot of hole	Clamp the workpiece to the bench vise	Bench vise (T-58)	
	Definition of spot coordinates	Center punch (T-43)	Keep the punch located spot in vertical direction
		Ball-peen hammer (T-66)	
	Remove the workpiece		
Create a hole	Clamp the workpiece to the table of the drill press	Vertical drilling machine (M-02)	
	Place the drill bit in the chuck	Twist drill (C-26)	Set the hole depth and cutting speed
	Place the drill bit in the chuck	Twist drill (C-36)	
	Remove the workpiece		
Create the screw threads	Clamp the workpiece to the bench vise	Bench vise (T-58)	
		Tap holder (T-45)	
	Place the tap in the holder	1st Taped tap (C-154)	Rotate two revolutions and turn back one-third cycle.
	Place the tap in the holder	2nd Taped tap (C-154-2)	
	Place the tap in the holder	3rd Taped tap (C-154-3)	
	Remove the workpiece		
			Browsing mode...

(*c*) Details of each task for screw threads generation appropriate manufacturing resource, is designated, and then planning processes are executed with the retrieved data from the DB. The tasks schedule model is subsequently stored in a DB that can be used by other users. Moreover, the implementation is based on the concept of the OOP paradigm. As an example, Fig. 14 shows the sequence diagram to build the tasks schedule model, which is represented by UML. A message is generated by user and sent to the client program to retrieve all information from the DB. Then, the chosen screw threads are again sent to the client program to get its cutter and tools. By a java class name **VR object operator**, the retrieved models are displayed on a browser where user interactions with a VR object occur. For example, the information of screw threads is used to get all available tools that are related to the screw threads. The message of tools and VR objects is sent to the **STGS construction** class to form a hierarchical screw threads generation structure by repeatedly analyzing the topological tools feasibility. Subsequently, the STGS is used to execute the tasks schedule with the tool selection. The following examples describe graphic user interfaces (GUIs) of the developed web-based VR applications in this paper. These GUIs consist of java applets and browsers.

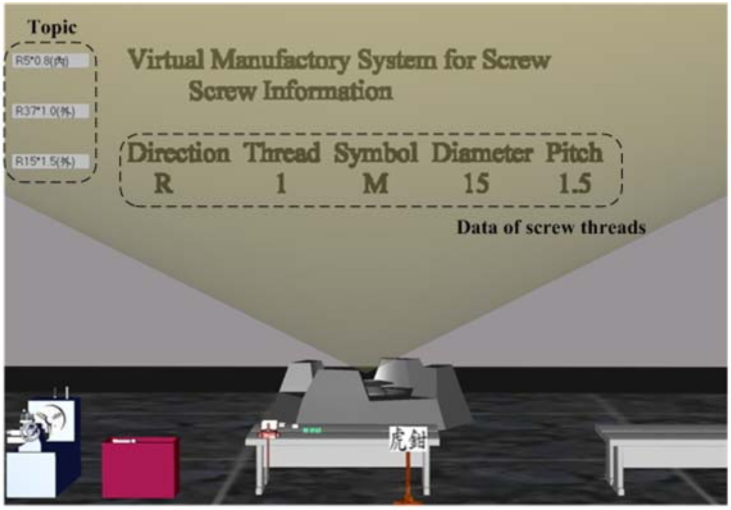
Fig. 15 continued.



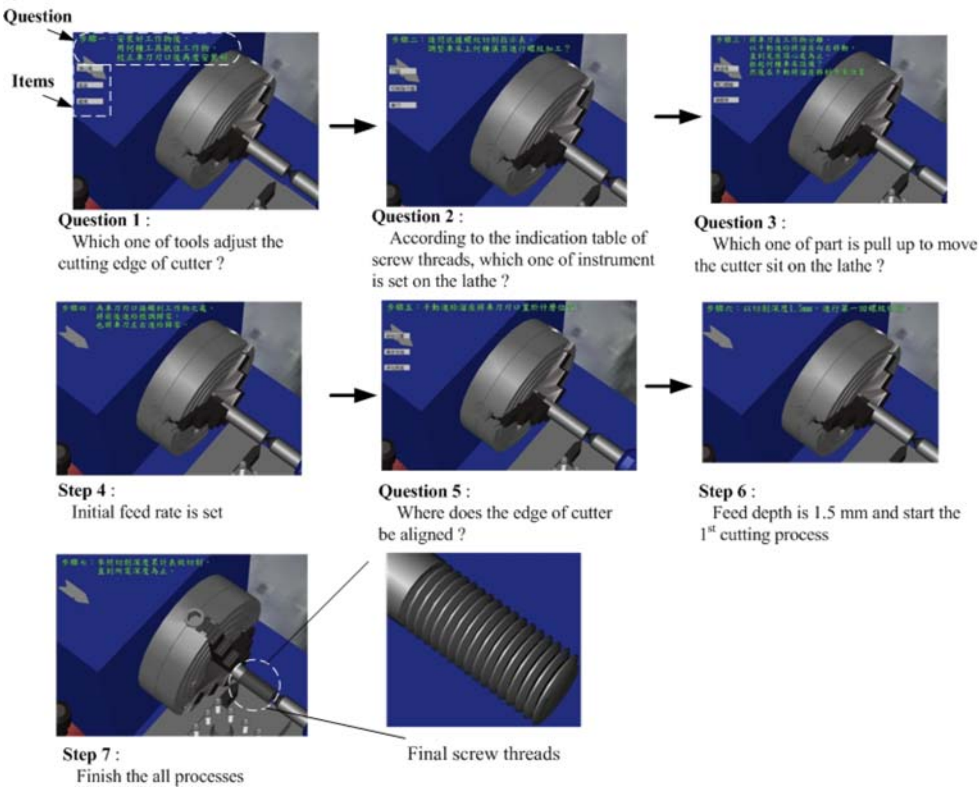
(*d*) Browsing mode for generating the screw threads

Example 1 is based on the machining of ISO metric general-purpose internal screw threads. As in shown Fig. 15a, when the user enters the Design Mode, a series of settings will be made using GUIs, which derive the form of screw threads of **R 1 M 10×1.0**. After information about tool list (including choice of tool, grips and fixtures, cutter and relevant parameters) is confirmed, the system will automatically generate a tasks schedule table that is easy to understand for user (as shown in Fig. 15b) except tasks schedule model. The table includes two sections—Tool List and Tasks Schedule of Screw Threads Generation—and they are created through Task-oriented decision approach. The former contains Information of Screw Threads and Tool List, where the content of the tool is based on user-input parameters. The latter lists the detailed machining process. To view the details of all tasks, simply select "Details" and look into Details of Task of Screw Threads Generation, as shown in Fig. 15c. The table illustrated in

Fig. 16 **a** Self-practice topics for screw threads generation. **b** Self-practice mode of external screw threads generation through the lathe



(*a*) Self-practice topics for screw threads generation



(*b*) Self-practice mode of external screw threads generation through the lathe

Fig. 15c contains Task name, Method, Tool, and Annotation, each of which are provided with actions to be taken and points of attention. The virtual machining process can also be viewed in Browsing Mode, in which each step of the machining process is displayed in correspondence to the tasks detailed in the tasks schedule table, as shown in Fig. 15d.

In practice thread machining, learner can choose Practice Mode from the main page shown in Fig. 15a and then select the practice topic from the menu on the left-hand side, as in Fig. 16a. This example uses ISO metric general-purpose external screw threads **R 1 M 15×1.5** for practice, and the lathe is used to cutting external thread. For each important step, several items are provided, as shown in Fig. 16b. Advance to next step is allowed when correct answers are given by user (single-choice type); otherwise, the system will return to the topic page.

5 Conclusion and further work

The virtual manufacturing environment over the internet can deal with a number of what-if scenarios for a product being designed. It is based on the real-time simulation and validation to minimize the uncertainty issues associated with a product development. In this paper, a user-friendly training platform has been developed that can tackle the design problem of screw threads and subsequent visualization of the screw threads generation process in a virtual environment. The user may have an overall idea about the screw threads to be manufactured because of the above information and procedure. Meanwhile, the proposed tasks schedule model follows the object-oriented modeling paradigm and uses four types of modules in this paper: the virtual component corresponding to the object model that describes the objects in the system and their relationships, the transfer operator corresponding to the functional model, which depicts the data transformations of the system, the state manager and flow controller corresponding to the dynamic model that represents the interactions among objects in the system.

There are some beneficial results offered in manufacturing engineering and other fields: (1) It presents an object-oriented methodology for the modeling and simulation of a virtual manufacturing environment; (2) VEs provide a means for users to navigate and investigate an efficient decision-making over entire process of manufacturing; (3) the internet technology provides a feasible tool for the data update and access timely; (4) manufacturing data can be retrieved through internet by integrating tasks schedule planning and the web interface; (5) to the implementation of the proposed tasks schedule model, the paper employs the OOP design, relational database management system, and the virtual reality technology. It provides an interaction between user and virtual components. The virtual platform generated can build the tasks schedule model and perform the precise time-dependent 3D simulation for machining screw threads; (6) virtual manufacturing of products and the digital mock-up of production scenarios gains a new quality due to the chosen concept and the consequential use of methods of virtual reality and multimedia. It is easily possible to generate setups of new products, test their functions, and plan new manufacturing facilities.

Because current DFX methods, e.g., design for manufacturing, are based on a static manufacturing environment and the islands of automation and uncertainty are caused by disregarding resources in real manufacturing conditions, further work on the integration of existing DFX tools with the developed internet-based virtual manufacturing environment and the evaluation of system efficiency for applications in a variety of manufacturing tasks is needed.

Acknowledgments Important parts of the system, especially the user interface and programming, were developed by the students Lu Yong-Xin and Lin Jun-Jia.

References

1. ISO metric screw threads (1998) ISO 68-1: ISO general purpose screw threads—basic profile—metric screw threads. International Organization for Standardization, Switzerland
2. ISO metric screw threads (1998) ISO 261: ISO general purpose metric screw threads—general plan. International Organization for Standardization, Switzerland
3. Unified thread standard (1995) ASME/ANSI B1.15—Unified Inch Screw Threads, UNJ Thread Form. ASME/ANSI in the United States
4. Unified thread standard (2003) ASME/ANSI B1.1—Unified Inch Screw Threads, UN & UNR Thread Form. ASME/ANSI in the United States
5. Chou C, Hsu HL, Yao YS (1997) Construction of a virtual reality learning environment for teaching structural analysis. Comput Appl Eng Educ 5(4):223–230
6. Johnson A, Roussos M, Leigh M, Vasilakis C, Barnes C and Moher T. The NICE project: learning together in a virtual world. Accessed on Feb. 2007. Accessed at http://www.ice.eecs.uic.edu/nice/nice/papers/vraishtml
7. Bolter J (1997) Virtual reality and the redefinition of self. In: Strate L, Jacobson R, Gibson S (eds) Communication and cyberspace–social interaction in an electronic environment. The Hampton, Cresskill, pp 105–119
8. Jern M (1997) Information visualization enables interaction with large datasets on the web. In: Earshaw R, Vince J (eds) The internet in 3D—information, images and interaction. Academic, San Diego, pp 223–236
9. Riva G, David F (2003) Communications through virtual technology: identity community and technology in the internet Age. IOS, Amsterdam
10. Ellis SR (1994) What are virtual environments? IEEE Computer Graphics & Application 14:17–22
11. Kalawsky RS (1993) The science of virtual reality and virtual environments. Addision-Wesley, Reading

12. Burdea G, Coiffet P (1994) Virtual reality technology. Wiley, New York
13. Krueger W, Frohlich B (1994) The responsative workbench (virtual work environment). IEEE Comput Graph Appl 14(3):12–15
14. Tesic R and Banerjee P (1999) Design of virtual objects for exact collision detection in virtual reality modeling of manufacturing processes. Proceedings of International conference on Robotics and Automation, Detroit, USA
15. Arangarasan R and Gadh R (2000) Geometric modeling and collaborative design in multimodel, virtual environment. Proceedings of ASME, IDETC/CIE Conference, Sept. 10–13
16. Roy S, Pohit G and Saha KN (2003) Computer aided design of spur gear. Proceedings of 20th AIMTDR Conference, BIT Mesra, Ranchi, India, Dec. 13–15
17. Pattanayak RK, Pohit G and Saha KN (2003) Application of solid modeling in virtual manufacturing of spur gear. Proceedings of 11th National Conference on Machines and Mechanism (Nacomm), I.I.T. Delhi, Delhi, Dec. 18–19, pp. 683–688
18. Lau YKH, Mak LK, Lu THM (2003) A virtual design platform for interactive product design and visualization. J Mater Process Technol 139(1–3):402–407
19. Qin FS, Harrison R, West AA, Wright KD (2004) Development of a novel 3D simulation modeling system for distributed manufacturing. Comput Ind 54(1):69–81
20. Seo Y, Kim D, Suh S (2006) Development of Web-based CAM system. Int J Adv Manuf Technol 28(11–12):101–108
21. Sherman WR, Craig AB (2003) Understanding virtual reality. Morgan Kaufmann, San Francisco
22. Pohit G (2005) Application of virtual manufacturing in generation of gears. Int J Adv Manuf Technol 31(1–2):85–91
23. Lambert AJD (2003) Disassembly sequencing: a survey. Int J Prod Res 41(16):3721–3759
24. Srinivasan H, Gadh R (2002) A non-interfering selective disassembly sequence for components with geometric constraints. IIE Trans 34:349–361
25. Ko H, Park MW and Lee HJ (2002) Conceptual framework of tangible space initiative and its application scenario to heritage alive. Proceedings of International Conference on Virtual Systems and Multimedia
26. Bierbaum A, Just C, Hartling P, Meinert K, Baker A and Cruz-Neira C (2001) VR juggler: a virtual platform for virtual reality application development. Proceedings of IEEE VR, pp 89–96
27. Liu X, Dodds G, McCartney J, Hinds BK (2004) Virtual DesignWorks—designing 3D CAD models via haptic interaction. Comput-Aided Des 36(3):1129–1140
28. Oakley I, Adams A, Brewster S, Gray P (2002) Guidelines for the design of haptic widgets. Proceedings of British HCI Conference, pp 195–212
29. Kuo CT (2000) Disassembly sequence and cost analysis for electronmechanical products. Robot Comput-Integr Manuf 16 (1):43–54
30. Yin PZ, Ding H, Li XH, Xiong LY (2003) A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. Comput Aided Des 35(1):37–56
31. Lazzerini B, Marcelloni F (2000) A genetic algorithm for generating optimal assembly plans. Artif Intell Eng 14(4):319–329
32. Soromaz D, Khosknevis B (1997) Machine and tool constraint specification for integrated process planning system. Proceedings of the 7th Industrial Engineering Research Conference, pp 901–906
33. Usher MJ, Fernandes JK (1999) An object-oriented application of tool selection in dynamic process planning. Int J Prod Res 37 (13):2879–2894
34. Fernandes JK, Raja HV (2000) Incorporated tool selection system using object technology. Int J Mach Tools Manuf 40(11):1547–1555
35. Cao Q, Dowlatshahi S (2005) The impact of alignment between virtual enterprise and information technology on business performance in an agile manufacturing environment. J Oper Manag 23 (5):531–550
36. Subramaniam V, Lee KD, Ramesh T, Hong SD, Wong SY (2002) Machine selection rules in a dynamic job shop. Int J Adv Manuf Technol 16(9–10):902–098
37. Moon C, Lee M, Seo Y, Lee YH (2002) Integrated machine tool selection and operation sequencing with capacity and precedence constraints using genetic algorithm. Comput Ind Eng 43(3):605–621
38. Choi BK, Kwan HH, Park TY (1996) Object-oriented graphical modeling of FMSs. Int J Flexible Manuf Sys 8(1):159–182
39. Blaha MR, Rumbaugh JR (2005) Object-oriented modeling and design with UML 2/e. Prentice Hall, Englewood Cliffs
40. Fishman GS (2001) Discrete-event simulation: modeling, programming and analysis. Springer, New York