

CREACIÓN DE LABORATORIOS VIRTUALES PARA LA FORMACIÓN PRÁCTICA EN INGENIERÍA DE FABRICACIÓN

Autores: Eva María Rubio Alvir*, Miguel Ángel Sebastián Pérez*, Alfredo Sanz Lobera**

Dirección: * Departamento de Ingeniería de Construcción y Fabricación

ETS Ingenieros Industriales de la UNED

Apartado de correos 60.149

28080-Madrid

** Departamento de Materiales y Producción Aeroespacial

ETS Ingenieros Aeronáuticos-UPM

E-mail: erubio@ind.uned.es

Resumen

En el presente trabajo se ha desarrollado una metodología que permite crear *entornos virtuales* con los que simular el comportamiento de equipos y procesos de fabricación ubicados en *entornos reales*. Dicha metodología se ha particularizado en el desarrollo de una Célula Flexible de Fabricación Virtual (CFFV), como primer paso hacia la creación de un Laboratorio Virtual de los Procesos de Mecanizado Automatizados.

Atendiendo al grado de interacción que le está permitido al usuario, se puede considerar la CFFV creada, dividida en dos fases. La primera es un *entorno virtual pasivo* con el que se puede observar la Célula en funcionamiento trabajando en modo desatendido, mientras que la segunda, de tipo *activo*, permite manipular sus principales componentes como si fueran objetos reales.

1. INTRODUCCIÓN

La enseñanza, dentro del ámbito de la Ingeniería, de ciertas áreas específicas tales como la Fabricación requiere etapas de formación práctica que ayuden a consolidar el aprendizaje teórico.

Su realización suele desarrollarse en laboratorios en los que se ven involucrados una serie de recursos materiales y humanos que en muchas ocasiones suponen unos desembolsos económicos mayores de los que puede soportar cualquier departamento universitario.

Además, la puesta en marcha de las máquinas y los dispositivos para la realización de las sesiones prácticas entraña ciertos riesgos, asociados a la manipulación de los mismos por personas no expertas, que pueden derivar en daños tanto personales como materiales.

En el caso particular de los Sistemas Flexibles de Fabricación (SFF); ambos problemas (inversiones y seguridad operativa) se agudizan, ya que son sistemas cuya implantación y puesta en marcha conlleva grandes gastos al estar constituidos por equipos de coste muy elevado (Máquinas-Herramienta de Control Numérico y Robots) .

Sin embargo, debido, principalmente, a que dichos sistemas permiten aumentar la productividad y la flexibilidad, son cada vez más numerosos dentro del sector industrial. Por esta razón, las empresas que los tienen instalados demandan a la sociedad personas con la preparación adecuada para la manipulación de los mismos.

Los Centros Educativos, ya sean de Formación Profesional o Universitarios, deben responder a esta demanda aportando a la sociedad egresados con la cualificación adecuada, en la programación y el manejo de MHCN y Robots integrados, para su inmediata incorporación al mundo del trabajo.

Para cubrir esta necesidad, la solución adoptada por algunos Centros Educativos, especialmente Universidades, es la creación de Laboratorios en los que se ha implantado algún pequeño Sistema Flexible de Fabricación, generalmente Células Flexibles de Fabricación (CFF), en los que los alumnos pueden completar su formación hasta conseguir la capacitación adecuada a la demanda del mercado y realizar trabajos de investigación que permitan mejorar u optimizar las implantaciones ya existentes.

No obstante, la solución es cara y poco asequible para algunos departamentos universitarios con escasos recursos económicos. Además, a pesar de ser máquinas de pequeños tamaños y potencias, existe el peligro de accidente durante su empleo, lo que supone un handicap adicional para su instauración.

Actualmente, la Realidad Virtual se presenta como uno de los mejores medios que existen para paliar en parte estos problemas; al permitir desarrollar laboratorios virtuales que resultan ser más económicos y seguros que los reales y que posibilitan llevar a cabo prácticas tanto presenciales como en línea.

Seguidamente, se muestran los pasos a seguir para crear un Laboratorio Virtual que pueda ser empleado en la realización de prácticas de los Sistemas Flexibles de Fabricación. Concretamente, se ha desarrollado una Célula Flexible de Fabricación Virtual que permite tanto la observación del funcionamiento global de este tipo de sistemas como la manipulación de los principales componentes que la forman.

2. DIRECTRICES GENERALES PARA LA CREACIÓN DE LABORATORIOS VIRTUALES

2.1. Definición de objetivos

La creación de cualquier aplicación de Realidad Virtual se inicia con la definición de los objetivos estableciendo una formulación conceptual del problema para, a partir de ella, ir sintetizando lo que se quiere conseguir con la aplicación. Para ello hay que: identificar y definir qué se va a recrear (*entorno real*), identificar las características de las personas a las que va a ir dirigida la aplicación y determinar la interrelación que va a existir entre los usuarios y el *entorno virtual* (interacción).

Una vez definidos los objetivos es necesario comprobar que existe una *f fuente de información* adecuada sobre el *entorno real* que se va a simular y que entre las personas que vayan a hacer uso de dicha información, haya alguna que cuente con la experiencia y con el *conocimiento* apropiado sobre el *entorno* a virtualizar. La necesidad de contar con alguien versado en el tema, que sea capaz de reflexionar, analizar y sintetizar la información recibida es vital para alcanzar el nivel de realismo requerido para la aplicación y para conseguir extraer las mejores conclusiones sobre los resultados obtenidos tras su ejecución. Además, es necesario que los *entes*, elementos del *entorno real*, se puedan simular mediante los *modelos*, componentes del *entorno virtual* que recogen la naturaleza dual de los *entes*, es decir, su parte física (*entidades*) y su parte conductual (*reglas de comportamiento*). Y, por último, es necesario que existan los medios adecuados, generalmente programas de software, para la construcción de *entidades* y *reglas de comportamiento*.

Posteriormente, hay que realizar un estudio de factibilidad en el que se analice si con los recursos humanos y materiales con los que se cuenta es posible crear la aplicación. Suele ser necesario tener que modificar los objetivos si las limitaciones existentes impidieran la obtención satisfactoria de los mismos.

Por último, si la amplitud o complejidad de los objetivos de la aplicación es elevada, puede ser necesario establecer fines parciales, en cuya definición habrá que seguir también los pasos expuestos.

2.2 Planificación del trabajo

Para la obtención de los objetivos propuestos, definidos en el primer punto, se proponen las tareas mostradas en la siguiente figura.

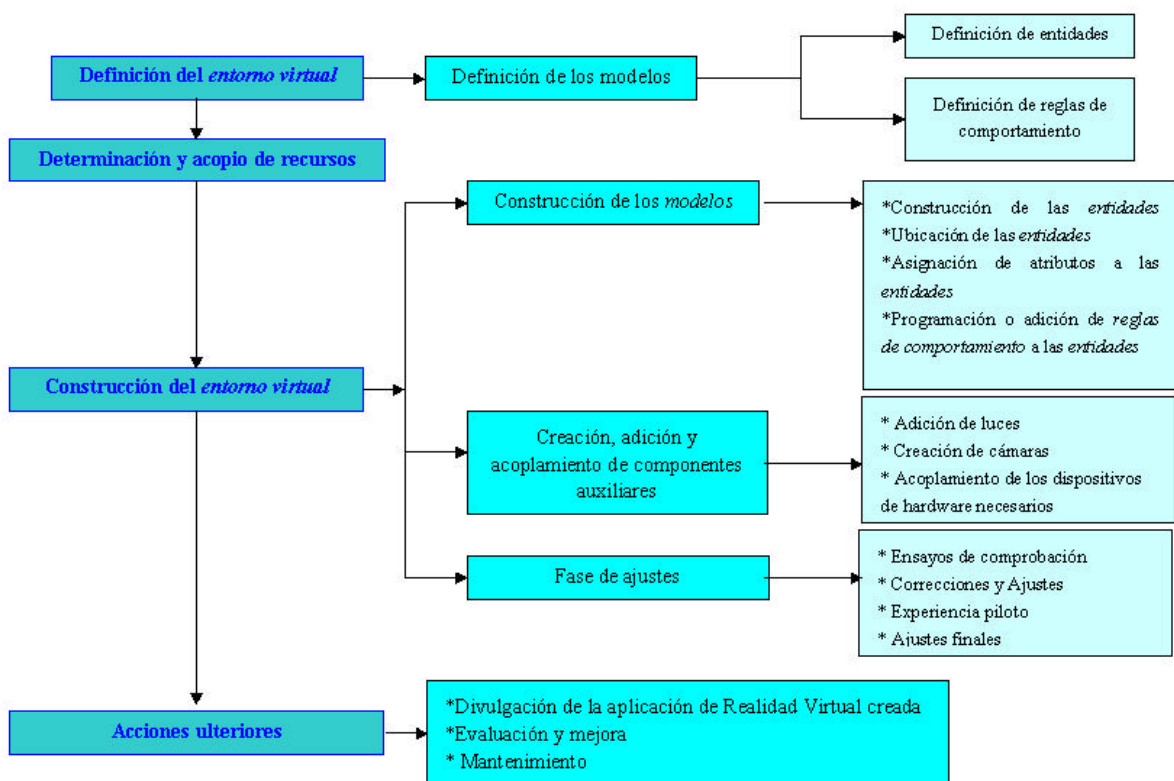


Figura 1. Esquema de planificación del trabajo.

2.3. Definición del entorno virtual

Una vez identificados, en la etapa de definición de objetivos, los *entes* del *entorno real* que se quieren recrear hay que definir los *modelos* que los representarán en el *entorno virtual*.

Los *modelos* van a estar constituidos por *entidades* y por *reglas de comportamiento*. Las primeras representan las características físicas de los *modelos* mientras que las segundas reproducen sus leyes de conducta o de actuación.

La definición física de los *modelos* implica determinar tantas *entidades* visuales, auditivas, hápticas u olfativas como estados puedan presentar los *modelos* en la evolución de la aplicación. Seguidamente se exponen algunas consideraciones sobre las *entidades* visuales que son las de uso más frecuente.

2.3.1. Definición de entidades

Las *entidades* visuales de los *modelos* suele realizarse por medio de polígonos [Burdea, 1994], [Stampe, 1994], [Banerjee, 2001]. Su principal inconveniente es que para lograr *entidades* con aspecto realista el número de polígonos debe ser elevado y como cada uno de ellos está representado por una posición y una orientación, en cuyo cálculo y almacenamiento se consume cierta cantidad de memoria, cuantos más polígonos haya, se necesitará más memoria y más tiempo para actualizar la pantalla; disminuyendo el nivel de fotogramas y perdiendo el realismo conseguido empleando *entidades* más perfeccionadas.

En la práctica, para aumentar la velocidad de actualización de la pantalla y optimizar los recursos de memoria disponibles por la máquina, suele ser habitual definir para cada uno de los *modelos* presentes en la aplicación varias *entidades* con distinto *nivel de detalle* y sustituir unas por otras en función de la distancia a la que se encuentren con respecto al usuario durante la evolución de la aplicación. También puede recurrirse a otras técnicas como son la radiosidad, el trazado de rayos o el mapeado de texturas.

2.3.2. Definición de *reglas de comportamiento*

En cuanto a la definición de *reglas de comportamiento* hay, en primer lugar, que identificar las principales interacciones que pueden llegar a darse entre los *entes* y entre éstos y el o los usuarios, en segundo lugar, hay que seleccionar de entre todas las interacciones identificadas aquéllas que se quieren simular y; por último, habrá que establecer los algoritmos que permitan recoger las interacciones y los comportamientos que harán posible constituir, finalmente, los *modelos* de la aplicación.

2.4. Determinación y acopio de recursos

Determinadas las *entidades* y las *reglas de comportamiento* de los *modelos* del *entorno virtual* que se han decidido simular, habrá que determinar qué recursos humanos y materiales son necesarios para poder conseguirlo y hacer acopio de los mismos.

En cuanto a los recursos humanos, lo ideal es contar con personas que tengan un adecuado *conocimiento* del *entorno* que se quiere recrear para poder lograr el éxito de la aplicación. Las limitaciones que van a afectar a la determinación del número de personas que será necesario disponer para desarrollar una aplicación concreta, van a venir impuestas por factores tales como, el plazo de entrega y los recursos económicos disponibles.

Los recursos materiales, principalmente el hardware y el software de modelización y de virtualización, se seleccionarán, mediante un estudio comparativo, de entre aquellos programas y dispositivos capaces de reproducir las características de las *entidades* y de las *reglas de comportamiento*. La elección final suele estar afectada por las mismas limitaciones vistas para los recursos humanos.

2.5. Construcción del Entorno Virtual

2.5.1. Bases generales

A la hora de abordar la construcción de cualquier *entorno virtual* suele ser conveniente tener en cuenta las siguientes recomendaciones [Burdea, 1994][Parra, 2001][Vince, 2001][Banerjee, 2001]:

- Dividir, si es posible, el *entorno virtual* en *escenas*, de modo que se exima al ordenador de representar las *entidades* de los *modelos* no presentes en la *escena* mostrada.
- Utilizar *entidades* simples con las que se tenga la ejecución fluida del *entorno virtual*.
- Utilizar *entidades* con distintos *niveles de detalle* que permitan optimizar los recursos del ordenador.
- Seleccionar las texturas empleadas (principalmente, las denominadas *texturas dinámicas* que representan, por ejemplo, el agua o la lluvia) ya que, consumen una gran cantidad de memoria.
- Definir pocas superficies curvas que emplean mucha memoria y producen el efecto *escalera*.
- Utilizar el criterio de baja resolución. porque ésta es costosa en tiempo y memoria.
- Emplear una iluminación heterogénea para potenciar la sensación de realismo.

- Usar sonidos sólo si son esenciales para lograr el realismo deseado; pues emplean mucha memoria.

2.5.2. Construcción de los *modelos*

El proceso de fabricación de un *entorno virtual* comienza con la construcción de las *entidades*. Generalmente, debido a la importancia que tiene el sentido de la vista, se empieza por la representación tridimensional de todas las *entidades* que intervendrán en la aplicación, para pasar después, en caso de que así se haya previsto en la definición física de los *modelos*, a incorporar otros efectos, como pueden ser, auditivos, hápticos e, incluso, olfativos (no obstante, existen aplicaciones desarrolladas para personas invidentes, en las que sólo hay que definir las características físicas de los *modelos* relativas al oído o al tacto [Barbero, 2001]).

Para una aplicación de Realidad Virtual típica (en la que el sentido de la visión es el más importante) se comienza por la importación al programa de software de Realidad Virtual de las *entidades* individuales o de conjuntos de *entidades* dispuestos según una determinada ubicación, creados previamente con el software de modelización 3D.

La construcción de la dimensión auditiva de las *entidades* es habitual conseguirla por medio de la Función de Transferencia de la Cabeza o HRTF, calculada por medio de la presión que produce en cada tímpano una fuente sonora. Esta función es extremadamente compleja de evaluar ya que hay que tener en cuenta factores externos (el azimut, la elevación y el alcance desde donde se produce el sonido o la frecuencia del mismo) e internos (los pabellones auditivos) al usuario [Rivero, 1999].

La construcción de las *entidades* hápticas requiere, en primer lugar, una serie de sensores de presión y temperatura sobre los puntos que se quieran estimular por medio del tacto y la fuerza. En segundo lugar, recoger todos los datos que sean precisos para reconstruir con el realismo deseado las sensaciones hápticas. Y, por último, hacen falta actuadores que sean capaces de emular dichas sensaciones.

En la construcción de las *entidades* olfativas se emplean los mismos métodos que en la elaboración de perfumes y esencias y son ofrecidas al usuario, posteriormente, por medio de sólidos, líquidos o geles.

Construidas todas las dimensiones físicas de los modelos es necesario colocar las visuales en la posición y con la orientación que deberán tener inicialmente en el *entorno virtual (condiciones iniciales)* y guardar esos datos relativos a su situación espacial en la memoria del ordenador.

Fijadas las *condiciones iniciales* hay que añadirles *atributos* para que adquieran ciertos rasgos propios o peculiaridades físicas que les permiten asemejarse aún más a los *entes* reales a los que representan (solidez o penetrabilidad, flexibilidad o rigidez, gravedad o ingravidez, opacidad o transparencia,...)

Incorporados los *atributos*, se ha de pasar a la programación propiamente dicha en la que se añadirán las *reglas de comportamiento* a las *entidades* hasta conformar los *modelos*. La programación va a poder efectuarse de diversas maneras: con librerías gráficas, motores 3D o suites de desarrollo.

2.5.3. Creación, adición y acoplamiento de componentes auxiliares

Para que la aplicación de Realidad Virtual desarrollada pueda ejecutarse correctamente suele ser habitual: tener que añadir luces, para que las *entidades* puedan verse; colocar cámaras virtuales para poder observar el *entorno virtual* desde distintos puntos de vista que de otra forma resultarían muy difíciles de contemplar; y acoplar los dispositivos de hardware correspondientes a las dimensiones sensoriales simuladas de manera que el usuario pueda percibir a través de ellos cuantas sensaciones visuales, auditivas, hápticas y olfativas se hubieran considerado idóneas en la etapa de definición de los objetivos.

2.5.4. Fase de ajustes

Según se va avanzando en la programación hay que realizar ensayos de comprobación que aseguren al programador que la integración de las *entidades* en grupos más grandes y complejos se va realizando correctamente. No hacerlo puede acarrear problemas mayores, ya que, repasar unas cuantas líneas de programación o enlaces entre unos pocos bloques de programación siempre es más fácil que revisar el programa completo. También pueden ir encaminados a comprobar si se han logrado las metas previstas en cuanto al grado de realismo conseguido.

Tras realizar los ensayos de comprobación el programador debe corregir los errores detectados en la fase anterior (si los hay); ajustar los parámetros de las *entidades* individuales para su perfecto funcionamiento

cuando pasan a formar parte de un conjunto; detectar la existencia de limitaciones en la creación del *entorno virtual*; y encontrar el equilibrio entre *lo que se quiere* y *lo que realmente se puede* conseguir.

Corregidos los posible errores detectados tras la fase de ensayos de comprobación, el programador debe poner la aplicación a disposición de usuarios “de confianza” con la idea de que se detecten los errores que aún pudieran existir y se aporten todo tipo de sugerencias.

Por último, habrá que incorporar a la aplicación las correcciones y las mejoras sugeridas en la etapa anterior, logrando realizar, de esta manera, los últimos ajustes antes de la primera publicación de aquélla.

2.6. Acciones ulteriores

Terminada la construcción del *entorno virtual*, se pondrá al alcance de los usuarios para los que ha sido ideado y realizado, a través de los medios que se consideren más efectivos.

Posteriormente, suele ser adecuado fijar un período de prueba tras el cuál se pedirá a los usuarios que rellenen un formulario de evaluación y que añadan cuantas sugerencias crean oportunas. De cuyo análisis se procurará el perfeccionamiento de la aplicación; incorporándole tantas mejoras como sea posible hasta conseguir un funcionamiento más refinado. Asimismo, será el momento de añadir las innovaciones aportadas por el avance de otras tecnologías.

En ocasiones, puede ser necesario establecer un plan de mantenimiento de la aplicación; sobre todo en aquellas en las que se emplean dispositivos de hardware específico ya que, puede ser necesario tener que realizar algún tipo de conservación o chequeo de los mecanismos integrantes de dichos dispositivos para comprobar que siguen ofreciendo las características requeridas por la aplicación.

3. LABORATORIOS VIRTUALES DE FABRICACIÓN

3.1. Planteamiento

Vistas las directrices generales para la elaboración de una aplicación de Realidad Virtual se va a mostrar ahora, como ejemplo, el desarrollo de un Laboratorio Virtual de los Sistemas Flexibles de Fabricación que pueda ser empleado en la realización de prácticas y que permita tanto la observación del funcionamiento global de este tipo de sistemas como la manipulación de sus principales componentes.

3.2. Definición de objetivos

En este caso se ha elegido como *entorno real* a simular mediante uno *virtual*, un Sistema Flexible de Fabricación, en concreto, una Célula Flexible de Fabricación; lo que permite simplificar el problema sin perder generalidad. Su principal objetivo es facilitar la comprensión del comportamiento individual y de conjunto de los componentes de la misma y cuyos principales usuarios serán, en principio, personas interesadas en temas relacionados con la Automatización de la Producción.

La aplicación se puede considerar dividida en dos fases: una de observación, en la que el usuario asiste a una escena en la que se puede ver una Célula Flexible de Fabricación funcionando en modo desatendido y otra de manipulación, en la que se le permite interactuar con los componentes de la misma, de modo similar a como lo haría con los equipos reales.

Por otra parte, como en este caso, en primer lugar, existe información sobre los componentes y los elementos auxiliares de los SFF que se encuentra recogida en la teoría de Control Numérico, de Máquinas-Herramienta y de los SFF, se puede afirmar que existe una fuente de información fiable sobre los SFF. En segundo lugar, los entes del entorno real, es decir, las máquinas-herramienta, los robots, las cintas transportadoras o los vehículos de transporte, por citar sólo algunos de ellos, se pueden representar en el entorno virtual mediante modelos y, por último, hay programas de software comerciales tanto para la creación de las entidades (3D Studio Max, Maya, etc.) como para la programación de las reglas de comportamiento (Virtools Dev, Sense 8, etc.) que hacen posible la creación de los modelos se puede concluir que el entorno real elegido se puede representar por medio de uno virtual.

Hecho esto, se ha de decidir qué *entes*, del *entorno real*, se desean representar, en concreto, se han tomado como elementos integrantes de la Célula Flexible de Fabricación Virtual: un robot, un centro de mecanizado y su panel de control, un ordenador que represente el sistema de control, un almacén de materias primas, otro de productos terminados, las piezas y los elementos auxiliares necesarios para la correcta colocación de los anteriores elementos.

Un amplio grupo de los usuarios serán alumnos de segundo y tercer ciclo de Ingeniería de todas aquellas especialidades en las que se impartan asignaturas relacionadas con la Fabricación Automatizada o que realicen actividades de investigación. Esto hace que, en principio, el perfil de los usuarios sea bastante homogéneo en cuanto a: edad, nivel intelectual y recursos propios con los que contarán para ejecutar la aplicación.

Comprobado que el *entorno real* se puede simular mediante un *entorno virtual* y perfilados los *entes* a simular y los condicionantes introducidos por el colectivo de usuarios a los que va dirigida la aplicación, se ha de realizar un estudio de factibilidad en el que se analicen si los recursos materiales y humanos disponibles serán o no suficientes para crear la aplicación. En este caso, se ha concluido que sí lo son al no requerirse medios especiales para llevar adelante la aplicación descrita.

3.3. Definición de los *modelos*

El *entorno real* a representar está formada por los elementos enumerados más arriba. De ellos, los más importantes son el robot y el centro de mecanizado con su panel de control; ya que, es sobre su aspecto y su comportamiento sobre lo que se quiere que los usuarios aprendan. El resto bastará con que se ajuste a unos estándares básicos.

Se ha considerado adecuado emplear sólo una *entidad* por cada *ente* del *entorno real* simulado y que éstas representen los elementos básicos de aquellos (sin tener en cuenta, tornillos, cables, etc.). También se ha creído oportuno definir un único *nivel de detalle* para cada *entidad* y dejar para las etapas de ensayos de comprobación y de ajustes, la posibilidad de incorporar nuevas *entidades* con *niveles de detalle* menores que el inicial en el caso de que los recursos del ordenador se saturen.



Figura 2. Scorboter VII y fresadora EMCO VMC 100.

Como *ente* real del robot y del centro de mecanizado se han tomado un Scorboter VII y una fresadora EMCO VMC 100 como los mostrados en la figura 2, de los que se conocen las principales características físicas y de actuación.

En cuanto a las *reglas de comportamiento* del robot Scorboter VII real, se puede decir que éste cuenta con cinco ejes sobre los que pueden girar las distintas partes que lo componen y abrir y cerrar la pinza que lleva en el extremo de su brazo. Cada uno de los movimientos descritos produce un sonido característico como consecuencia de la puesta en marcha de los motores que mueven las articulaciones.

En este caso, se han seleccionado: el movimiento de las distintas partes del robot alrededor de los ejes mencionados y la apertura y el cierre de la pinza.

Por su parte, el centro de mecanizado EMCO VMC 100 real puede realizar las siguientes actuaciones: abrir y cerrar la puerta, abrir y cerrar la mordaza, desplazar la mordaza para acercarla o alejarla de la herramienta que se encuentra activa durante el mecanizado, seleccionar la herramienta que trabajará durante el mecanizado mediante el giro del portaherramientas, girar la herramienta activa y añadir refrigerante durante el mecanizado. Además, cada una de estas actuaciones produce sonidos característicos siendo los más importantes el de la puesta en marcha de la herramienta y el del mecanizado.

Para el centro de mecanizado se han seleccionado como acciones a reproducir, abrir y cerrar la puerta, abrir y cerrar la mordaza, desplazar la mordaza para acercarla o alejarla de la herramienta activa durante el mecanizado, girar la herramienta activa y el ruido del mecanizado y el del giro de la herramienta.

Los algoritmos a definir serán distintos en cada una de las fases de la aplicación. Así, en la primera fase en la que el observador ha de ver una Célula Flexible de Fabricación Virtual funcionando sola como si estuviese controlada desde su sistema de control, cada una de las acciones que realicen el robot y el centro de mecanizado, deberá ser accionada por algún medio interno del programa utilizado. Mientras que en la segunda fase, en la que es el observador el que las pone en marcha habrá que establecer los medios para la entrada de datos al programa de modo que se activen las distintas acciones programadas.

3.4. Determinación y acopio de recursos

En este caso no es necesario emplear hardware específico pues las *reglas de comportamiento* elegidas no lo requieren. Además, si en un futuro se elabora material multimedia para que los usuarios lo ejecuten con sus propios medios, tampoco sería conveniente emplearlo; ya que, éstos dispondrán, por lo general de teclado y ratón como medios para introducir datos e interactuar con el *entorno* y todos los esfuerzos realizados en conseguir acoplar dispositivos de hardware específico serían baldíos.

En lo que respecta a la selección del software de programación, se ha elegido la *suite de desarrollo* Virtools Dev 2.0 que puede ser utilizada por programadores y por no-programadores; característica que la convierte en una herramienta de trabajo muy versátil. Cuenta con la ventaja adicional de ser compatible con diversos programas de modelización de *entidades* tales como Maya o 3D Studio Max, programas que pueden importar y exportar gran variedad de tipos de ficheros y que permiten, de este modo, a Virtools Dev quedar abierto a esa misma posibilidad. Además, los requerimientos que demanda en cuanto al hardware básico y al software necesario para su instalación son bastante asequibles en relación al mercado actual.

3.5. Construcción del *entorno virtual*

3.5.1. Construcción de los *modelos*

Teniendo en cuenta las consideraciones previas a la construcción de cualquier *entorno virtual* vistas en el punto 2.5.1., se ha abordado la construcción de la presente aplicación. En primer lugar, se ha comenzado por importar, al programa de software de virtualización, las *entidades* creadas con el de modelización. En segundo lugar, para simular el ruido producido por el giro de la herramienta en vacío y por el mecanizado de las piezas, se han utilizado los sonidos (ficheros *.wav) con los que Virtools Dev 2.0 cuenta entre sus recursos. Por último, teniendo en cuenta los objetivos planteados en esta aplicación, no ha sido necesario construir las dimensiones háptica y olfativa puesto que no han sido consideradas.

Importadas las *entidades* visuales, se ha procedido a colocarlas en la posición y con la orientación que han de tener inicialmente en el *entorno virtual*, fijando esas condiciones como *condiciones iniciales*. Además se han asignado los siguientes *atributos*: al suelo, el atributo *floor* que hará que cualquier otra

entidad que se halle dentro de su campo de acción caiga y permanezca sobre él cuando se le asigne el comportamiento de *keep on floor*. Al resto de las *entidades* que intervienen se les han asignado los *atributos fixed obstacle* y *moving obstacle* según sean obstáculos fijos o móviles, para poder detectar las colisiones que pueden darse entre ellos a lo largo de la evolución del *entorno virtual*.

La programación se realiza, básicamente, uniendo entre sí mediante enlaces (*links*), bloques programados previamente (*Building Blocks*) que, al ser activados, realizan una tarea sobre la *entidad* a la que están asociados.

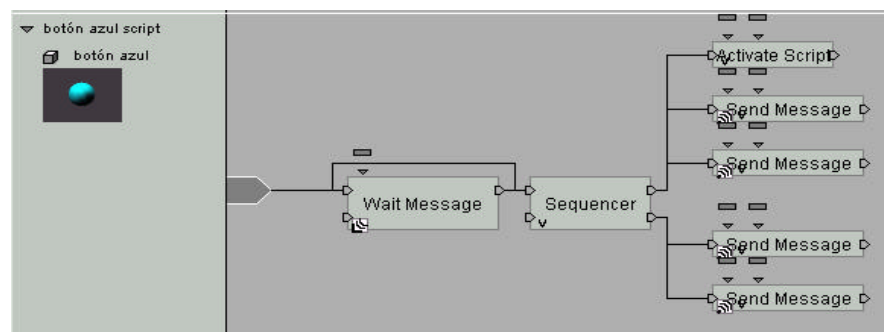


Figura 3. Ejemplo de unión de *Building Blocks*.

3.5.2. Creación, adición y acoplamiento de componentes auxiliares

Las características particulares de Virtools Dev hacen necesaria la adición de luces nada más importar las *entidades* al software de virtualización, para que pueda verse la escena. Además, hay que colocar cámaras virtuales para observar ciertos detalles del funcionamiento de la célula y de sus dispositivos, que no se podrían ver de otra manera. Y, por último, como se ha creído oportuno no utilizar dispositivos de hardware específicos de Realidad Virtual no ha sido necesario su acoplamiento.

3.5.3. Fase de ajustes

Por las características propias de Virtools Dev ha sido necesario llevar a cabo un riguroso y sistemático proceso de comprobación de la programación; lo que ha supuesto realizar las etapas de: ensayos, correcciones y ajustes, simultáneamente con aquella.

Obtenida la primera versión de la aplicación se puso a disposición de un reducido grupo de personas para que la sometieran a cuantos ensayos y pruebas consideraran convenientes, tras lo cuál, se realizaron los ajustes finales

3.6. Acciones ulteriores

La difusión de la aplicación puede realizarse bien, en espacios reservados para tal fin (como pueden ser los laboratorios de Fabricación de la UNED y de la UPM, donde se ha realizado este estudio) puestos a disposición de los usuarios o, bien, por medio de CD-ROM o de Internet para que puedan tener acceso a la misma, con sus propios medios.

La mejora de la aplicación se hará en base al análisis de los comentarios, críticas y sugerencias recibidos por parte de los usuarios después de un cierto tiempo de uso, aunque con las acciones llevadas a cabo tras la experiencia piloto, se puede decir que se ha conseguido ya un primer nivel de mejora.

Por último, debido a que en este caso no se han empleado dispositivos de hardware específico, no se considera necesario tener que establecer un plan de mantenimiento para la aplicación.

4. CONCLUSIONES

Se ha detectado la necesidad de desarrollar nuevas vías para la realización de prácticas en Ingeniería de Fabricación más económicas y seguras que las tradicionalmente empleadas por algunas universidades.

En el presente trabajo intentando dar respuesta a dicha necesidad se ha establecido una metodología para la creación de Laboratorios Virtuales sobre los que poder llevar a cabo la formación práctica. En concreto, se ha desarrollado un *entorno virtual* que representa un Sistema Flexible de Fabricación y que permite tanto la observación como la manipulación de sus elementos

El desarrollo de la aplicación ha permitido, además, sentar las bases para la creación de Laboratorios Virtuales de SFF más complejos así como de otros tipos de Procesos de Fabricación diferentes.

5. AGRADECIMIENTOS

Los autores quieren agradecer a la Comisión Interministerial de Ciencia y Tecnología española la financiación del proyecto de investigación DPI00-1419 en cuyo contexto ha sido realizado el presente trabajo.

6. REFERENCIAS

1. Banerjee, P.; Zetu, D. (2001). *Virtual Manufacturing*. John Wiley & Sons, Inc. New York.
2. Barbero, J.I. (2001). *Computer graphics access for blind people through a haptic virtual environment*. VR World Congress. Universitat Pompeu Fabra. Barcelona.
3. Burdea, G.C.; Coiffet, P. (1994). *Virtual Reality Technology*. John Wiley & Sons, Inc. New York.
4. Mc Carthy, C.; Callede, D. (2001). *Virtools Dev User Guide*. Virtools. París.
5. Parra, J.C.; García, R.; Santelices, I. (2001). *Introducción Práctica a la Realidad Virtual*. Ediciones Universidad de Bío-Bío. Concepción. Chile. <http://zeus.dci.ubiobio.cl/~sigradi/libros/>
6. Rivero, N. (1999). *Audio 3D. Diseño de Sistemas basados en DPS*. Facultad de Ingeniería. Universidad de la República. Uruguay. <http://www.iie.edu.uy/~nicolas/dsp1999/index.html>
7. Rubio, E.M. (2002). *Aplicación de Técnicas de Realidad Virtual al estudio del comportamiento de Sistemas Flexibles de Fabricación*. Tesis Doctoral. UNED. Madrid.
8. Rubio Alvir, E.M.; Sebastián Pérez, M.A.; Sanz Lobera, A. (2001). *Desarrollos para el apoyo y mejora de la docencia del Control Numérico*. Los educadores ante el reto de las tecnologías de la información y la comunicación. III Congreso Internacional de Informática Educativa. Madrid.
9. Sanz, A.; Rubio, E.M.; Sebastián, M.A. (2002). *Considerations to the application of Virtual Reality techniques for the study and the training with Numerical Control of Machine-Tools*. CIMEC-2002. CIRP International Manufacturing Education Conference. University of Twente. Enschede (The Netherlands).
10. Stampe, D.; Roehl, B.; Eagan, J. (1994). *Realidad Virtual. Creaciones y Desarrollo*. Ediciones Anaya Multimedia, S.A. Madrid.
11. Vince, J. (2001). *Essential Virtual Reality Fast*. Springer. London.

[VOLVER AL INDICE TEMAS](#)