

Especificación y modelaje de diseños en CAD

Rodrigo Cardoso

Olga Lucía Cardoso

José Tiberio Hernández

Oscar Ramos

e-mail: rcardoso@andescol.bitnet

jhermand@andescol.bitnet

dfac@andescol.bitnet

Grupo DFAC

Departamento de Ingeniería de Sistemas y Computación

Universidad de los Andes

Bogotá, D.E., Colombia

Resumen

Se presenta un marco conceptual dentro del que se define formalmente la dinámica del proceso de diseño. En él puede abordarse con claridad la implantación de ambientes de diseño apoyados computacionalmente (CADE). Los ambientes que resultan son interactivos, adaptivos, de diseño descendente, y su implantación parece natural en el paradigma de programación orientada por objetos.

Palabras claves

CAD, CAM, diseño, bosquejo, rasgo, diseño descendente, diseño adaptivo, reutilización, programación orientada por objetos.

1 INTRODUCCION

Los sistemas actuales de CAD/CAM son fundamentalmente herramientas de ayuda en la documentación y validación de diseños terminados. El proceso mismo de diseño carece de herramientas de apoyo y debe adelantarse sin mayores ayudas computacionales. Lo usual es introducir el diseño al sistema, validar algunos aspectos mediante herramientas de análisis y aprovechar la informática para su eventual reutilización ([EID90],[Tom87]).

Siendo 'objetos terminados', los diseños se centran en una representación geométrica enriquecida con una descripción de atributos no gráficos. Las herramientas CAD pretenden una genericidad capaz de ser útil en todos los dominios imaginables.

El proceso real de diseño es una aproximación gradual a la especificación completa de un objeto que satisfaga requerimientos impuestos desde un principio o aun dentro del proceso en sí. El objeto terminado es el elemento final de una secuencia de bosquejos, en la que los detalles definitivos se establecen gradualmente. Durante el proceso se elaboran y sopesan alternativas de bosquejos, y se toman decisiones que determinen el rumbo definitivo del diseño. Por otra parte, el diseño es una labor multidisciplinaria en la que cooperan personas que estudian diferentes aspectos, que deben integrarse armónicamente tanto en los bosquejos como en el producto final [You92].

Para solucionar las deficiencias mencionadas de los sistemas actuales, de suerte que verdaderamente se apoye el proceso de diseño, proponemos un marco conceptual en el que se puedan definir sistemas CAD que llenen los vacíos anotados. Más precisamente, recomendamos el uso de sistemas interactivos de apoyo al diseño adaptivo descendente, que puedan sintonizarse a dominios específicos.

La sección 2 establece conceptos básicos que permiten entender con precisión la labor de diseño, sus partes y su desarrollo. En la sección 3 se comentan, de manera general, aspectos que deben ser tenidos en cuenta al enfrentar la implantación de ambientes de diseño que se apoyen en las ideas de 2. Se señala el paradigma orientado por objetos (OO) como la plataforma natural para llevar a cabo dicha implantación [Tom89]. En la sección 4 se resumen conclusiones relevantes. Deliberadamente se ha excluido -en aras de la brevedad- la ilustración, mediante ejemplos, de muchos de los conceptos tratados. Los autores atenderán gustosamente cualquier inquietud o aclaración que se estime necesaria.

2 CONCEPTOS BASICOS

Llamaremos *producto* a una descripción completa del objeto que se pretende diseñar. Todos los atributos y funciones relevantes y sus interrelaciones deben ser precisamente establecidos. Adicionalmente, se desea que los productos sean "fabricables", i.e., realizables en el contexto de un ambiente de fabricación entendido.

Un *bosquejo* es una descripción incompleta de un producto. Entre bosquejos se puede definir una relación de orden \leq , con el significado de "menos definido que", i.e., $b_1 \leq b_2$ establece que el bosquejo b_1 es menos detallado que el bosquejo b_2 . Menos detallado puede querer decir, que en b_2 hay algún atributo que en b_1 no está definido, o lo está parcialmente.

Un bosquejo b *representa* todos los productos que puedan ser igual o más definidos que él. Este conjunto se llamará el *dominio* de b , y se denotará $dom(b)$. El conjunto de los bosquejos se denominará *Bosquejos*.

El (*proceso de*) *diseño* se entenderá como un grafo finito dirigido acíclico en el que los nodos son bosquejos y los arcos reflejan la relación de orden \leq . Un producto es, entonces, un nodo terminal de un diseño, que sea fabricable. El conjunto de los diseños se denotará *Diseños*.

Antes de diseñar (i.e. construir diseños), se entiende que se ha establecido un conjunto de *requerimientos* que un producto deberá satisfacer. En el mismo proceso del diseño puede ser necesario refinar o detallar en algún sentido lo requerido.

2.1 Derivación de diseños

El trabajo del diseñador consiste en construir diseños a partir de requerimientos. Comienza con un diseño *vacío* (i.e. un grafo vacío), al que se van agregando bosquejos y arcos entre ellos. La decisión de agregar un nodo depende de los requerimientos iniciales así como de las restricciones que puedan resultar de otras decisiones que se hayan debido tomar anteriormente.

Los nodos fuente en un diseño -a los que no llegan arcos- son bosquejos *primitivos*. Son conocidos para el diseñador, en el sentido de que no se discute su procedencia ni su construcción. Un bosquejo primitivo puede ser un tipo básico usualmente disponible en un lenguaje de programación, v.gr. números, cadenas de caracteres, listas, pilas, etc. En un dominio de diseño específico resulta de gran utilidad el disponer de bosquejos primitivos más elaborados que los ya mencionados. Por ejemplo, en el diseño de sistemas hidráulicos, resultaría útil contar con bosquejos primitivos como "bomba", e incluso "bomba rotacional" (y dentro de ésta: "bomba centrífuga", "bomba axial", etc.).

Entre los bosquejos primitivos se considera el bosquejo *vacío*. Con él se representan todos los productos posiblemente diseñables.

A partir de un diseño dado D , se puede construir un diseño *derivado* D' , agregando un nuevo nodo a D . Además de la posibilidad de agregar un bosquejo primitivo, es factible añadir un bosquejo más definido (i.e. con más detalles, atributos o funciones) que alguno de los de D . Así, el diseño derivado D' puede, también, considerarse como "más definido" que D . En este caso, se dice que el nuevo bosquejo agregado es *complejo*, y su definición exacta depende de la de sus predecesores en el grafo.

En el caso general, sean b_1, \dots, b_r bosquejos del diseño dado D , y b el bosquejo que se desea agregar, de manera que $b_1 < b$, para $i=1, \dots, r$. Se consideran tres posibilidades para llevar a cabo este proceso de *derivación*:

- *Enriquecimiento*: En este caso, $r=1$. El nodo b representa un bosquejo en el que se puede calcular una función f que no era considerada en el bosquejo b_1 . Las demás funciones que operaban sobre b_1 son heredadas por b . Para este caso, se dirá que $b = enr(b_1, f)$, y además $dcm(b) = dcm(b_1)$.
- *Composición*: En este caso, $r > 1$. El nodo b representa un bosquejo que junta todos los atributos considerados en los bosquejos b_1, \dots, b_r , que son llamados sus *componentes*. Se dirá que un *rasgo* (inglés: *feature*) de un bosquejo es una componente, o bien, una componente de un rasgo. Las funciones que operaban sobre cada b_i deben extenderse de manera que ahora operen sobre b . Para este caso, se dirá que $b = comp(b_1, \dots, b_r)$, y además $dcm(b) = dcm(b_1) \times \dots \times dcm(b_r)$.
- *Restricción*: En este caso, $r=1$. El nodo b representa un bosquejo en el que se exige que uno o más atributos ya considerados en el bosquejo b_1 satisfagan una relación p . Las funciones que operaban sobre b_1 son heredadas por b . Para este caso, se dirá que $b = rest(b_1, p)$, y además $dcm(b) = \{u \in dcm(b_1) \mid p(u)\}$.

Hay una forma especial de restricción, denominada *conexión*. En ésta, el bosquejo b_1 es el resultado de una composición, v.gr. $b_1 = comp(a_1, \dots, a_n)$. La relación de restricción p se refiere exactamente a una de las componentes, digamos a_k . Además, se quiere que $p(a_k)$ sea satisfecha por exactamente un elemento. Este único elemento debe ser un bosquejo primitivo fabricable, y se dice de él que es una *componente conectada*. Como notación, se dirá que $b = conx(b_1, k, p)$.

La notación introducida permite referirse a los bosquejos por su forma de construcción en términos de otros bosquejos menos definidos. En particular, los productos tienen como referencia una cadena de derivación que refleja la forma en que pueden llegar a realizarse.

2.2 Vistas de un diseño

En problemas complejos, en los que es necesaria la intervención de un grupo de diseñadores especialistas en diferentes áreas, o simplemente en los que un diseñador único desee dividir su trabajo según ciertos criterios, surge el concepto de *vista* de un bosquejo, y, por extensión, del diseño.

Una vista es una función $v: \text{Bosquejos} \rightarrow \text{Bosquejos}$ que "extrae" algunos aspectos "interesantes" de los bosquejos. Por ejemplo [Män90], si se tiene un bosquejo de una lámpara, una vista interesante puede ser su estructura, otra, su sistema eléctrico, etc. Nuestra definición de *vista* es aun más exigente: se desea poder explicar con precisión el significado de "extraer" aspectos interesantes.

Si v es una vista, el hecho de que $v(a)=a'$ significa que a' tiene algunos de los rasgos de a . Además, las funciones de a , cuyos dominios se formen exclusivamente con rasgos extractados, son heredadas por a' . Obsérvese que, el bosquejo a' resulta estructuralmente similar al bosquejo a ; de hecho, los dos bosquejos son iguales, salvo los aspectos olvidados en la extracción. La noción de vista se puede extender al conjunto de diseños, i.e. como función $v: \text{Diseños} \rightarrow \text{Diseños}$. Si $v(d)=d'$, d' es un diseño análogo a d , en el que se han extractado los aspectos relevantes de cada uno de los bosquejos que conforman a d ¹.

3 AMBIENTES DE DISEÑO DE PROPOSITO ESPECIFICO

El marco conceptual expuesto en la sección anterior permite plantear características generales, deseables en ambientes de diseño, asistidos por computador. Llamaremos CADE a un tal ambiente (inglés: *Computer Aided Design Environment*). Un CADE haría las veces de un administrador de diseños, permitiendo su evolución en la forma ya anotada. Tal funcionalidad incluiría, entre otras labores: validación de bosquejos, derivación de diseños, administración e integración de vistas, propagación de restricciones, reutilización de bosquejos, etc.

Es apenas natural esperar que un CADE sea *interactivo*. La actividad de diseñar es esencialmente una labor de 'edición' de bosquejos y diseños, que debe ser apoyada por el ambiente de manera flexible y amable para el usuario. Por otra parte, un CADE debe ser *adaptivo*, i.e. debe tener la posibilidad de apoyarse en bosquejos o diseños existentes para adecuarlos a otras situaciones, dependientes de los requerimientos en cuestión. En otras palabras, debe *reutilizar* soluciones ya conocidas.

La definición de diseño, como grafo dirigido de bosquejos, cuyos arcos significan "igual o más definido que", realiza una concepción inherentemente *descendente* (inglés: *top-down*) de la labor de diseño. La derivación de un diseño se apoya en la definición de otro menos definido. Este modo de trabajo no excluye la posibilidad de considerar, dentro de un diseño, alternativas que permitan explorar distintas maneras de satisfacer los requerimientos dados. En particular, es posible considerar, a partir de un bosquejo no terminal, formas de derivación diferentes de las ya establecidas. Es decir, las acciones de *reintento* (inglés: *backtracking*) pueden adelantarse naturalmente.

Un CADE debería administrar *versiones* de bosquejos, de manera que el diseñador pudiera comparar cómodamente alternativas. Esta característica es particularmente deseable cuando pueden concebirse varios productos-versiones, y debe elegirse uno de ellos para ser fabricado [Mar90].

¹ Algebraicamente se puede definir, en forma natural, un homomorfismo $v': \text{Bosquejo} \rightarrow \text{Bosquejo}$, tal que, si $v(a)=a'$, entonces $v'(a')=a$. La extensión de vistas a diseños hace factible la extensión de v' a un homomorfismo de grafos.

Los bosquejos primitivos que sirven de comienzo en la derivación de diseños pueden ser de propósito general (v. gr., el bosquejo vacío, los tipos de un lenguaje de programación). No obstante, si sólo éstos fueran permitidos, la reutilización de soluciones ya conocidas distaría mucho de ser verdaderamente automatizable, y muchos diseños deberían "reinventar la rueda". Por tanto, es deseable que un CADE manipule bases de bosquejos primitivos, en donde resida el conocimiento que refleje la experiencia de los diseñadores.

Más aun, un CADE de propósito general, i.e., en el que pudiera diseñarse cualquier producto, parecería demasiado ambicioso para ser considerado como una buena herramienta. Un CADE de *propósito específico* serviría en la elaboración de diseños para dominios más pequeños y precisos. Así, además de disponer de bases de bosquejos primitivos adecuados a las posibles necesidades, podría ser factible operar los bosquejos/diseños de maneras quizás especializadas, distintas de las consideradas en 2.

3.1 Hacia una implantación de un CADE

En primera instancia, habría que pronunciarse sobre la plataforma tecnológica en la que podría implantarse un CADE. Las definiciones y consideraciones ya mencionadas parecen apuntar en forma natural hacia el tratamiento del problema de diseñar con herramientas *orientadas por objeto* [Tom89]. En este sentido, se explicará en forma breve cómo los conceptos definidos pueden rephrasearse en términos como clases, objetos, métodos, herencia, etc., de forma que su manipulación también resulte natural en este contexto.

Un bosquejo puede representarse por una clase. La relación $.> .$ (más definido que) corresponde a la relación de "ser subclase de"². El dominio de un bosquejo es el conjunto de todas sus posibles instancias, como clase. La clase universal representa el bosquejo vacío, y un producto es una clase que sólo tiene una instancia posible.

Los atributos y funciones de bosquejos son atributos y métodos de las clases correspondientes. En un producto están "totalmente implantados", i.e., corresponden a objetos predefinidos y operaciones ejecutables en la plataforma computacional de que se disponga. En un bosquejo que no sea producto se tendrán "implantaciones parciales", i.e., los atributos o métodos están definidos de manera paramétrica³, de suerte que la instanciación de los parámetros con productos correspondientes resulte en una instancia que sea, a su vez, un producto.

² Se está entendiendo que la clase b' es subclase de la clase b cuando la definición de b' incluye todos los atributos y métodos definidos para b .

³ En la terminología de objetos, se habla de *genericidad*, y su implantación se logra a través de las llamadas *clases abstractas*.

Los bosquejos primitivos corresponden a clases predefinidas o disponibles al comenzar la tarea de diseñar. Las operaciones de derivación consisten en construir una clase (bosquejo) b a partir de la definición ya disponible de otras clases:

- *Enriquecimiento*: Si $b = enr(b_1, f)$, entonces la clase b es una subclase b_1 , a la que se aumenta la definición del método f . Este puede ser genérico.
- *Composición*: Si $b = comp(b_1, \dots, b_r)$, la clase b hereda los atributos y métodos de las clases b_1, \dots, b_r .
- *Restricción*: Si $b = rest(b_1, p)$, la clase b hereda los atributos y métodos de la clase b_1 , pero sólo son instancias de b aquéllas de b_1 que satisfagan el predicado de restricción p . En la práctica, el predicado p debe ser expresable y calculable en el ambiente de programación disponible.

El control de un CADE así implantado debería administrar los requerimientos que deben satisfacerse, tanto los inicialmente planteados, como los que surjan a lo largo del proceso. Por ejemplo, es importante que cualquier bosquejo que sea definible sea en algún sentido realizable, i.e. debe poder exhibirse un producto que sea instancia de él. Una manera de garantizar la *consistencia* del diseño podría considerar exigir del usuario la definición de un ejemplo que satisficiera las condiciones que se imponen sobre un bosquejo, en cada derivación por restricción.

El manejo de *alternativas* aparece al permitir que un bosquejo tenga más de un sucesor posible. Las variantes que así surgen son, en el supuesto de que los requerimientos sean cabalmente atendidos, soluciones diferentes para el problema planteado. La selección de una mejor puede depender de criterios adicionales, considerables quizás sólo al final del proceso.

3.2 Manipulación de vistas

Extendida a diseños, como se sugiere en 2.2, la noción de vista sirve para entender, separadamente, diferentes aspectos del diseño de un producto. Esta cualidad parece plausible, en especial cuando el diseño ya se entiende como un todo, acabado. Sin embargo, los beneficios que deberían derivarse de la posibilidad de definir vistas tendrían que ser capitalizables *durante el proceso* de diseño. Así, un CADE que maneje vistas deberá coordinar, dinámicamente, la evolución paralela de los procesos parciales de diseño que representan diferentes vistas de un solo diseño.

Hay problemas de coordinación que pueden presentarse, análogos a los que pueden existir en la manipulación de información que resulta significativa para varios usuarios, que la utilizan de manera simultánea⁴. De hecho, dos vistas pueden compartir rasgos de un bosquejo, y si una de ellas impone condiciones sobre estos rasgos (v.gr. los

⁴ Este es el problema de la coherencia de la información de una base de datos con usuarios que pueden efectuar transacciones de modificación, en principio simultáneamente.

define, los compara, etc.), la otra debería conocer tales restricciones y verificar su factibilidad. Naturalmente, debería asegurarse que las dos vistas no intentan modificar o restringir, al mismo tiempo, los rasgos comunes.

Así, si se considera un *diseño global*, definible, en algún sentido, como la unión de varias vistas, un CADE debe tener en cuenta que

- el diseño global debe reflejar los cambios en los diseños vistas,
- los diseños vistas deben conocer las condiciones sobre rasgos que los afecten, que sean causados por otras vistas,
- la consideración de alternativas en diseños vistas conlleva la consideración de alternativas en el diseño global y en otros diseños vistas,
- una vista puede vetar una alternativa -quizás sugerida por otra vista- si no se puede encontrar un producto que realice, en el contexto de la primera vista, el bosquejo resultado de considerar las nuevas restricciones.

La implantación de mecanismos de control y coordinación que brinden los servicios de definición de alternativas y vistas tiene como principal dificultad la *propagación de restricciones* entre vistas, así como la explosión combinatoria del número de bosquejos-alternativas que resultan de considerar variantes en varios bosquejos-*vista* que compartan rasgos de definición. Para intentar remediar o aliviar estas dificultades, se debería, en lo posible:

- definir vistas independientes, i.e. que no compartan rasgos,
- limitar la definición de alternativas, dentro de diseños-*vistas*. O bien, considerar alternativas "internamente", dentro de cada vista, y elegir el menor número posible de ellas, antes de comunicarlas a otras vistas con rasgos comunes, a través de los mecanismos de concertación que en este caso sean necesarios.

3.3 Sobres de presentación

La reutilización de bosquejos se constituye en la forma, por excelencia, de ahorrar esfuerzos y costos dentro del proceso de diseño. Un CADE debe administrar almacenes de bosquejos, que puedan ser consultados y reutilizados por los usuarios, de manera cómoda [Ram91].

En muchas ocasiones, el diseñador no necesita enterarse, en todo posible detalle, de las características de los productos que un bosquejo describe. En realidad, interesan mucho los detalles de interrelación con el exterior (v.gr. costo, número de conectores, dimensiones, disipación de calor, etc.), y poco o nada, los de la constitución interna (materiales, componentes internos, etc.).

Para un bosquejo, un *sobre de presentación* es una vista que extrae rasgos relevantes para la reutilización del bosquejo en diseños que no pretendan explicar el funcionamiento interno de los correspondientes productos. Por ejemplo, los productos que puedan ser conectados, y que por tanto no deben explicarse ni entenderse en su funcionamiento interior, deben tener sobres de presentación que mencionen únicamente sus atributos relevantes "para el exterior".

Un CADE debería contar con facilidades de consulta, manipulación y almacenamiento de bosquejos, y proveer estos servicios a través de un manejo eficaz de un *catálogo de sobres de presentación*.

3.4 Ambientes de propósito específico

Aunque se recomienda que un CADE se entienda especialmente útil para elaborar diseños dentro de un dominio específico, hay aspectos genéricos a cualquier CADE concebible, que pueden pensarse como constituyentes de un CADE *núcleo*. Este debe contar con bosquejos básicos análogos a los tipos de un lenguaje de programación (números, cadenas de caracteres, etc.). Todos los mecanismos de derivación de diseños, así como los servicios de edición de bosquejos (crear, guardar, etc.), deben ser disponibles.

En la concepción de un CADE de propósito específico debe adelantarse una labor de profesionalización de los servicios al área particular en que se desea diseñar. Así, un CADE orientado a resolver los problemas de un dominio específico resultaría de completar el CADE núcleo, logrando su especialización a través de

- establecer jerarquías de clasificación de bosquejos por rasgos relevantes al dominio,
- extender los servicios del núcleo, para manipulación de bosquejos especializada al dominio,
- considerar bosquejos primitivos especiales al dominio,
- definir y manejar vistas propias del dominio.

4 CONCLUSIONES

Se presentó un marco de ideas donde pudieron definirse conceptos relativos a la actividad del diseño -producto, bosquejo, diseño, rasgo, vista, versión-. Una óptica formal de tales nociones da sustento a la construcción de verdaderos ambientes de diseño apoyados por herramientas computacionales (CADE). Es importante resaltar que la labor de diseñar se ha formalizado pretendiendo modelar formas usuales de trabajo, de modo que se atienda, además del producto, al proceso de diseño como tal.

El enfoque utilizado pretende integrar aspectos expuestos en los modelos orientados por rasgos (inglés: *feature oriented*) de [Sre88], de ambientes CAD inteligentes e interactivos de [Vee89], de diseño descendente [Män90] y de ingeniería concurrente [You92]. Algunas características son motivadas por la praxis y otras por ideas propuestas, principalmente en [Yos83].

El paradigma OO se ha sugerido como plataforma tecnológica para el desarrollo de CADEs. En él se han rephraseado los conceptos generales, que incluso esbozan ideas sobre la definición de un CADE núcleo, que sirva de piedra angular para la construcción de otros, de propósito particular. Por lo demás, la realización de CADEs debe dar luces que orienten la construcción de otros, y esto debe ser particularmente cierto al intentar teorizar sobre la mejor forma de administrar la reutilización de bosquejos en dominios específicos.

El tratamiento formal y el manejo unívoco de los conceptos considerados señalan pautas claras para adelantar dentro de ellas la labor de implantación de CADEs, que actualmente se encuentran en curso en el laboratorio del grupo DFAC.

BIBLIOGRAFIA

- [Car91] Cardoso, R., *Verificación y desarrollo de programas*, Ediciones Uniandes, Bogotá, 1991.
- [dIR91] de la Rosa, F., Hernández, J.T., CAD de propósito específico: una propuesta por rasgos, Memo de Investigación, CIFI, Universidad de los Andes, Bogotá, agosto 1991.
- [EID89] El Dashan, K., Barthes, J.P., Implementing Constraint Propagation in Mechanical CAD Systems, en *Intelligent CAD Systems II*, Akman, V., ten Hagen, P.J.W., Veerkamp, P.J. (eds.), Springer Verlag, 1989.
- [Her92] Hernández, J.T., Rodríguez, M., Gutiérrez, E., Ferreira, A.C., Zendron, P.C.D., IAGE: A feature oriented CAPP System, 6th Conference on CAD/CAM, Robotics and Factories of the Future, Bera, H., Gill, R. (eds.), Southbank Press, 1992.
- [Män90] Mäntylä, M., A modeling system for top-down design of assembled products, IBM Journal of Research and Development, Vol. 34, No. 5, Septiembre 1990.
- [Mar90] Marín, R., Diseño conceptual adaptivo en sistemas mecánicos, Tesis M.Sc., Universidad de los Andes, Bogotá, abril 1990.
- [Ram91] Ramos, O., Hernández, J.T., Ambientes de diseño y modelaje orientado por rasgos, Memo de Investigación, CIFI, Universidad de los Andes, Bogotá, agosto 1991.
- [Sre88] Sreevalsan, S.P., Rogers, M.T., Billo, R., Mathew, A., *Current status of feature technology*, 2nd. edition, CAM-I Report R-88-GM-04, Arlington, 1988.
- [Suh90] Suh, N.P., *The principles of Design*, Oxford University Press, New York, 1990.
- [Tom87] Tomiyama, T., ten Hagen P.J.W., The Concept of Intelligent Integrated Interactive CAD Systems, CWI Report, Centre for Mathematics and Computer Science, Amsterdam, abril 1987.
- [Tom89] Tomiyama, T., Object Oriented Programming Paradigm for Intelligent CAD Systems, en *Intelligent CAD Systemems II*, Akman, V., ten Hagen, P.J.W., Veerkamp, P.J. (eds.), Springer Verlag, 1989.
- [Vee89] Veerkamp, P., Akman, V., Bernus, P., ten Hagen, P., IDDL: A Language for Intelligent Interactive CAD Systems, en *Intelligent CAD Systems II*, Akman, V., ten Hagen, P.J.W., Veerkamp, P.J. (eds.), Springer Verlag, 1989.
- [Yos83] Yoshikawa, H., CAD framework guided by general description theory, en *Proceedings of the IFIP WG 5.2 Working Conference on CAD systems framework*, North-Holland, Roros, 1983.
- [You92] Young, R.E., O'Grady, P., Greef, A., A Concurrent Engineering System: The Use of Constraint Networks, 6th Conference on CAD/CAM, Robotics and Factories of the Future, Bera, H., Gill, R. (eds.), Southbank Press, 1992.