



Estándar IEEE para la información Tecnología—Diseño de sistemas— Descripciones de diseño de software

Sociedad de Computación IEEE

Patrocinado por el
Comité de Normas de Ingeniería de Software y Sistemas

1016™

IEEE
Avenida Park 3
Nueva York, NY 10016-5997, EE. UU.

20 de julio de 2009

Norma IEEE 1016™-2009
(Revisión de
Norma IEEE 1016-1998)

Estándar IEEE para la información Tecnología—Diseño de sistemas— Descripciones de diseño de software

Patrocinador

Comité de Normas de Ingeniería de Software y Sistemas de la

Sociedad de Computación IEEE

Aprobado el 19 de marzo de 2009

Junta de Normas IEEE-SA

Resumen: Se describen el contenido y la organización de la información necesarios para las descripciones de diseño de software (SDD). Una SDD es una representación de un diseño de software que se utilizará para comunicar información de diseño a las partes interesadas. Se especifican los requisitos para los lenguajes de diseño (anotaciones y otros esquemas de representación) que se utilizarán para las SDD conformes.

Esta norma es aplicable a bases de datos automatizadas y lenguajes de descripción de diseño, pero puede utilizarse para documentos en papel y otros medios de descripción.

Palabras clave: preocupación por el diseño, tema del diseño, visión del diseño, punto de vista del diseño, diagrama, diseño de software, descripción del diseño de software

El Instituto de Ingenieros Eléctricos y Electrónicos, Inc.
3 Park Avenue, Nueva York, NY 10016-5997, EE. UU.

Copyright © 2009 del Instituto de Ingenieros Eléctricos y Electrónicos, Inc.
Todos los derechos reservados. Publicado el 20 de julio de 2009. Impreso en los Estados Unidos de América.

IEEE es una marca registrada en la Oficina de Patentes y Marcas de EE. UU., propiedad del Instituto de Ingenieros Eléctricos y Electrónicos, Incorporated.

PDF: ISBN 978-0-7381-5925-6 Norma estándar 95917
Versión impresa: ISBN 978-0-7381-5926-3 STDPD95917

Ninguna parte de esta publicación podrá reproducirse en ninguna forma, en un sistema de recuperación electrónica o de otro modo, sin el permiso previo por escrito del editor.

Los documentos de las Normas IEEE se desarrollan en el marco de las Sociedades IEEE y los Comités de Coordinación de Normas de la Junta de Normas de la Asociación de Normas IEEE (IEEE-SA). El IEEE desarrolla sus normas a través de un proceso de desarrollo de consenso, aprobado por el Instituto Nacional Estadounidense de Normas, que reúne a voluntarios que representan diversos puntos de vista e intereses para lograr el producto final. Los voluntarios no son necesariamente miembros del Instituto y prestan sus servicios sin remuneración. Si bien el IEEE administra el proceso y establece reglas para promover la imparcialidad en el proceso de desarrollo de consenso, el IEEE no evalúa, prueba ni verifica de forma independiente la precisión de la información ni la solidez de los juicios contenidos en sus normas.

El uso de una norma IEEE es completamente voluntario. La IEEE no asume ninguna responsabilidad por daños personales, materiales o de otro tipo, de cualquier naturaleza, ya sean especiales, indirectos, consecuentes o compensatorios, que resulten directa o indirectamente de la publicación, el uso o la confianza depositada en este documento o en cualquier otro documento de norma IEEE.

El IEEE no garantiza ni declara la exactitud o el contenido del material incluido en este documento y rechaza expresamente cualquier garantía expresa o implícita, incluida cualquier garantía implícita de comerciabilidad o idoneidad para un fin específico, o de que el uso del material incluido en este documento esté libre de infracciones de patentes. Los documentos de las normas IEEE se suministran "TAL CUAL".

La existencia de una Norma IEEE no implica que no existan otras formas de producir, probar, medir, comprar, comercializar o proporcionar otros bienes y servicios relacionados con el alcance de la Norma IEEE. Además, el punto de vista expresado en el momento en que se aprueba y emite una norma está sujeto a cambios provocados por los avances en el estado de la técnica y los comentarios recibidos de los usuarios de la norma. Toda Norma IEEE está sujeta a revisión al menos cada cinco años para su revisión o reafirmación, o cada diez años para su estabilización. Cuando un documento tiene más de cinco años y no ha sido reafirmado, o más de diez años y no ha sido estabilizado, es razonable concluir que su contenido, aunque todavía tiene algún valor, no refleja totalmente el estado actual de la técnica. Se advierte a los usuarios que comprueben que tienen la última edición de cualquier Norma IEEE.

Al publicar y poner a disposición este documento, el IEEE no sugiere ni presta servicios profesionales o de otro tipo para ninguna persona o entidad ni en su nombre. El IEEE tampoco se compromete a cumplir con ningún deber que otra persona o entidad tenga con otra. Cualquier persona que utilice este documento y cualquier otro documento de Normas IEEE debe confiar en su criterio independiente y ejercer un cuidado razonable en cualquier circunstancia dada o, según corresponda, buscar el asesoramiento de un profesional competente para determinar la idoneidad de una norma IEEE determinada.

Interpretaciones: Ocasionalmente pueden surgir preguntas sobre el significado de partes de las normas en relación con aplicaciones específicas. Cuando se le plantea al IEEE la necesidad de interpretaciones, el Instituto iniciará las acciones necesarias para preparar las respuestas adecuadas. Dado que las normas del IEEE representan un consenso de intereses, es importante asegurarse de que cualquier interpretación también haya recibido la concurrencia de un equilibrio de intereses. Por esta razón, el IEEE y los miembros de sus sociedades y comités coordinadores de normas no pueden proporcionar una respuesta instantánea a las solicitudes de interpretación, excepto en aquellos casos en que el asunto haya recibido previamente una consideración formal.

Una declaración, escrita u oral, que no se procese de acuerdo con el Manual de Operaciones del Consejo de Normas IEEE-SA no se considerará la posición oficial del IEEE ni de ninguno de sus comités y no se considerará ni se confiará en ella como una interpretación formal del IEEE. En conferencias, simposios, seminarios o cursos educativos, una persona que presente información sobre las normas IEEE deberá dejar en claro que sus puntos de vista deben considerarse los puntos de vista personales de esa persona y no la posición, explicación o interpretación formal del IEEE.

Los comentarios para la revisión de las normas IEEE son bienvenidos por parte de cualquier parte interesada, independientemente de su afiliación a la IEEE. Las sugerencias para cambios en los documentos deben presentarse en forma de una propuesta de cambio de texto, junto con los comentarios de apoyo correspondientes. Las recomendaciones para cambiar el estado de una norma estabilizada deben incluir una justificación de por qué se requiere una revisión o retiro. Los comentarios y recomendaciones sobre las normas, y las solicitudes de interpretaciones, deben dirigirse a:

Secretario, Consejo de Normas IEEE-SA

445 carril de las azadas

Piscataway, Nueva Jersey 08854

ciervo

El Instituto de Ingenieros Eléctricos y Electrónicos, Inc. otorga autorización para fotocopiar partes de cualquier norma individual para uso interno o personal, siempre que se pague la tarifa correspondiente al Copyright Clearance Center.

Para coordinar el pago de la tarifa de licencia, comuníquese con Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. También se puede obtener permiso para fotocopiar partes de cualquier estándar individual para uso educativo en el aula a través del Copyright Clearance Center.

Introducción

Esta introducción no es parte de IEEE Std 1016-2009, Estándar IEEE para Tecnología de la Información—Diseño de Sistemas—
Descripciones de diseño de software.

Esta norma especifica los requisitos sobre el contenido y la organización de la información para las descripciones de diseño de software (SDD). Una SDD es una representación de un diseño de software que se utilizará para registrar información de diseño, abordar diversas cuestiones de diseño y comunicar dicha información a las partes interesadas del diseño.

Los SDD desempeñan un papel fundamental en el desarrollo y mantenimiento de los sistemas de software. Durante su vida útil, los SDD son utilizados por compradores, gerentes de proyectos, personal de control de calidad, gerentes de configuración, diseñadores de software, programadores, probadores y encargados del mantenimiento. Cada una de estas partes interesadas tiene necesidades únicas, tanto en términos de información de diseño requerida como de organización óptima de esa información. Por lo tanto, una descripción de diseño contiene la información de diseño que necesitan esas partes interesadas.

La norma también especifica requisitos sobre los lenguajes de diseño que se utilizarán al producir SDD que cumplan con estos requisitos de contenido y organización.

La norma especifica que un SDD debe organizarse en una serie de vistas de diseño. Cada vista aborda un conjunto específico de preocupaciones de diseño de las partes interesadas. Cada vista de diseño está prescrita por un punto de vista de diseño. Un punto de vista identifica las preocupaciones de diseño en las que se centrará su vista y selecciona los lenguajes de diseño utilizados para registrar esa vista de diseño. La norma establece un conjunto común de puntos de vista para las vistas de diseño, como punto de partida para la preparación de un SDD, y una capacidad genérica para definir nuevos puntos de vista de diseño, ampliando así la expresividad de un SDD para sus partes interesadas.

Esta norma está destinada a utilizarse en situaciones de diseño en las que se debe preparar una descripción de diseño explícita. Estas situaciones incluyen actividades de diseño y construcción de software tradicionales que conducen a una implementación, así como situaciones de “ingeniería inversa” en las que se debe recuperar una descripción de diseño a partir de una implementación existente.

Esta norma se puede aplicar a software comercial, científico, militar y de otros tipos. Su aplicabilidad no está restringida por el tamaño, la complejidad o la criticidad del software. Esta norma tiene en cuenta tanto el software como su contexto de sistema, incluido el entorno de desarrollo y operativo. Se puede utilizar cuando el software comprende el sistema o cuando el software forma parte de un sistema más grande caracterizado por componentes de hardware, software y humanos y sus interfaces.

Esta norma es aplicable tanto si la SDD se captura mediante documentos en papel, bases de datos automatizadas, herramientas de desarrollo de software u otros medios. Esta norma no admite explícitamente, ni se limita a, el uso con ninguna metodología de diseño de software en particular o lenguajes de diseño en particular, aunque establece requisitos mínimos para la selección de esos lenguajes de diseño.

Esta norma se puede utilizar con IEEE Std 12207™-2008 [B21]; también se puede utilizar en otros contextos de ciclo de vida.

Esta norma consta de cinco cláusulas, como sigue:

La cláusula 1 define el alcance y el propósito de la norma.

La cláusula 2 proporciona definiciones de términos utilizados en el contexto de la norma.

^a Los números entre paréntesis corresponden a los de la Bibliografía del Anexo A.

La cláusula 3 proporciona un marco para comprender los SDD en el contexto de su preparación y uso.

La cláusula 4 describe el contenido y la organización requeridos de un SDD.

La cláusula 5 define varios puntos de vista de diseño para utilizar en la producción de SDD.

El Anexo A proporciona una bibliografía.

El Anexo B define cómo debe describirse de manera uniforme un lenguaje de diseño que se utilizará en un SDD.

El Anexo C contiene una plantilla para organizar un SDD conforme a los requisitos de esta norma.

Esta norma sigue el Manual de estilo de normas IEEE. ^b En particular, la palabra deberá identificar los requisitos que deben cumplirse para poder afirmar la conformidad con esta norma. El verbo debería identificar recomendaciones y el verbo puede utilizarse para indicar que determinados cursos de acción son permisibles.

Esta revisión de la norma se basa en la norma IEEE Std 1471™-2000 [B20] y amplía los conceptos de vista, punto de vista, parte interesada y preocupación de esa norma para respaldar el diseño y la construcción de alto nivel y detallados para software. La demarcación entre arquitectura, diseño de alto nivel y detallado varía de un sistema a otro y está fuera del alcance de esta norma.

Aviso a los usuarios

Leyes y reglamentos

Los usuarios de estos documentos deben consultar todas las leyes y reglamentaciones aplicables. El cumplimiento de las disposiciones de esta norma no implica el cumplimiento de ningún requisito reglamentario aplicable.

Los implementadores de la norma son responsables de observar o hacer referencia a los requisitos reglamentarios aplicables. El IEEE no pretende, mediante la publicación de sus normas, instar a que se tomen medidas que no cumplan con las leyes aplicables, y estos documentos no deben interpretarse como si lo hicieran.

Derechos de autor

Este documento está protegido por derechos de autor del IEEE. Se pone a disposición para una amplia variedad de usos públicos y privados. Estos incluyen tanto el uso como referencia en leyes y reglamentos, como el uso en la autorregulación privada, la normalización y la promoción de prácticas y métodos de ingeniería. Al poner este documento a disposición para su uso y adopción por parte de autoridades públicas y usuarios privados, el IEEE no renuncia a ningún derecho de propiedad intelectual sobre este documento.

^b

El Manual de estilo de normas IEEE se puede encontrar en <http://standards.ieee.org/guides/style/index.html>.

Actualización de documentos IEEE

Los usuarios de las normas IEEE deben tener en cuenta que estos documentos pueden ser reemplazados en cualquier momento por la publicación de nuevas ediciones o pueden ser modificados de vez en cuando mediante la publicación de enmiendas, correcciones o erratas. Un documento oficial del IEEE en cualquier momento consiste en la edición actual del documento junto con las enmiendas, correcciones o erratas vigentes en ese momento. Para determinar si un documento determinado es la edición actual y si ha sido modificado mediante la publicación de enmiendas, correcciones o erratas, visite el sitio web de la IEEE Standards Association en <http://ieeexplore.ieee.org/xpl/standards.jsp>, o comuníquese con el IEEE en la dirección indicada anteriormente.

Para obtener más información sobre la IEEE Standards Association o el proceso de desarrollo de estándares IEEE, visite el sitio web de IEEE-SA en <http://standards.ieee.org>.

Errata

Las erratas, si las hubiera, de esta y todas las demás normas se pueden consultar en la siguiente URL: <http://standards.ieee.org/reading/ieee/updates/errata/index.html>. Se recomienda a los usuarios que consulten periódicamente esta URL para comprobar si hay erratas.

Interpretaciones

Se puede acceder a las interpretaciones actuales en la siguiente URL: <http://standards.ieee.org/reading/ieee/interp/index.html>.

Patentes

Se llama la atención sobre la posibilidad de que la implementación de esta norma pueda requerir el uso de materia protegida por derechos de patente. Con la publicación de esta norma, no se toma ninguna posición con respecto a la existencia o validez de ningún derecho de patente en relación con la misma. El IEEE no es responsable de identificar las reivindicaciones de patentes esenciales para las que se pueda requerir una licencia, de realizar investigaciones sobre la validez legal o el alcance de las reivindicaciones de patentes o de determinar si los términos o condiciones de licencia proporcionados en relación con la presentación de una carta de garantía, si los hubiera, o en cualquier acuerdo de licencia son razonables o no discriminatorios. Se advierte expresamente a los usuarios de esta norma que la determinación de la validez de cualquier derecho de patente y el riesgo de infracción de dichos derechos es de su exclusiva responsabilidad. Se puede obtener más información de la IEEE Standards Association.

Participantes

En el momento en que esta norma se presentó a la Junta de Normas IEEE-SA para su aprobación, el Diseño de Software El Grupo de Trabajo de Descripciones estuvo integrado por los siguientes miembros:

Keith R. Middleton, presidente (en funciones)
Vladan V. Jovanovic, presidente (emérito)
Basil A. Sherlund, presidente (emérito)
Rich Hilliard, secretario y editor técnico

Guillermo Bartolomé
Edward Byrne
Bob cocinero

Ed Corlett
Philippe Kruchten

Tierra de Kathy
James Moore
Ira Sachs

Los siguientes miembros del comité de votación individual votaron sobre esta norma. Los votantes pueden haber votado a favor, en contra o abstención.

Butch Anton
Angela Anuszewski Chris
Bagge Pieter
Botman Juan
Carreon
Lawrence Catchpole
Danila Chernetsov
Keith Chow S.
Claassen
Daniel Conte
David Cornejo
Geoffrey Darnton
Terry Dietz
Antonio Doria
Timothy Ehrler
Kameshwar Eranki
Andre Fournier Eva
Freund David
Frischia Lewis
Gray Michael
Grimley Randall
Groves John Harauz

Mark Henley
Rutger A. Heunks Ian
Hilliard Werner
Hoelzl Peter Hung
Atsushi Ito
Mark Jaeger
Piotr Karocki
Dwayne Knirk
George Kyle Susan
Land Dewitt
Latimer David
J. Leciston Daniel
Lindberg Vincent
Lipsio Edward McCall
Keith R. Middleton
William Milam

James Moore
Rajesh Murthy
Michael S. Newman
William Petit
Ulrich Pohl

Robert Robinson
Randall Safier
James Sanders
Bartien Sayogo
Robert Schaaf
David J. Schultz
Stephen Schwarm
Raymond Senechal
Luca Spotorno
Thomas Starai
Walter Struppler
Gerald Stueve
Marcy Stutzman K.
Subrahmanyam Richard
Thayer John
Thywissen Thomas
Tullia Vincent
Tume John Walz
P. Wolfgang
Forrest Wright
Janusz Zalewski
Wenhao Zhu

Cuando el Consejo de Normas IEEE-SA aprobó esta norma el 19 de marzo de 2009, contaba con los siguientes miembros:

	Robert M. Grow, presidente	
	Thomas Prevost, vicepresidente	
	Steve M. Mills, expresidente	
	Judith Gorman, Secretaria	
Juan Barr	Alexander Gelman Jim	Ley de David
Karen Bartleson	Hughes Rich	Ted Olsen
Victor Berman	Hulett Young	Glenn Parsons
Bolsa de Ted	Kyun Kim Joseph L.	Ron Petersen
Richard De Blasio	Koepfinger* John Kulick	Narayanan Ramachandran
Andy Drozd		Jon Rosdahl
Marcos Epstein		Sam Sciacca

*Miembro Emérito

También se incluyen los siguientes enlaces sin derecho a voto del Consejo de Normas IEEE-SA:

Howard Wolfman, representante de TAB
Michael Janezic, representante del NIST
Satish Aggarwal, representante del NRC

Lisa Perry
Gerente del programa de normas IEEE, desarrollo de documentos

Hora de Malia
Gerente del programa de normas IEEE, Desarrollo de programas técnicos

Contenido

1. Descripción general	1
1.1 Ámbito de aplicación	1
1.2 Finalidad	2
1.3 Público al que va dirigido.....	2
1.4 Conformidad	2
2. Definiciones.....	2
3. Modelo conceptual para descripciones de diseño de software	3
3.1 Diseño de software en contexto.....	3
3.2 Descripciones del diseño de software dentro del ciclo de vida.....	6
4. Contenido de la información de descripción del diseño.....	7
4.1 Introducción	7
4.2 Identificación de SDD.....	8
4.3 Las partes interesadas en el diseño y sus preocupaciones.....	8
4.4 Vistas de diseño.....	8
4.5 Puntos de vista del diseño	9
4.6 Elementos de diseño.....	9
4.7 Superposiciones de diseño	11
4.8 Fundamento del diseño.....	12
4.9 Lenguajes de diseño.....	12
5. Puntos de vista del diseño	13
5.1 Introducción	13
5.2 Punto de vista del contexto.....	14
5.3 Punto de vista de la composición.....	15
5.4 Punto de vista lógico.....	16
5.5 Punto de vista de la dependencia	17
5.6 Punto de vista de la información	17
5.7 Los patrones utilizan el punto de vista	18
5.8 Punto de vista de la interfaz.....	19
5.9 Punto de vista de la estructura.....	20
5.10 Punto de vista de la interacción.....	20
5.11 Punto de vista de la dinámica de estados.....	21
5.12 Punto de vista del algoritmo.....	21
5.13 Punto de vista de los recursos	22
Anexo A (informativo) Bibliografía	24
Anexo B (informativo) Descripción del lenguaje de diseño conforme.....	26
Anexo C (informativo) Plantillas para un SDD.....	28

Estándar IEEE para la información Tecnología—Diseño de sistemas— Descripciones de diseño de software

AVISO IMPORTANTE: Esta norma no tiene como objetivo garantizar la seguridad, la protección, la salud o la protección del medio ambiente en todas las circunstancias. Los implementadores de la norma son responsables de determinar las prácticas o los requisitos reglamentarios adecuados en materia de seguridad, protección, protección, salud y medio ambiente.

Este documento IEEE está disponible para su uso sujeto a avisos importantes y exenciones de responsabilidad legales. Estos avisos y exenciones de responsabilidad aparecen en todas las publicaciones que contienen este documento y se pueden encontrar bajo el encabezado "Aviso importante" o "Avisos importantes y exenciones de responsabilidad sobre documentos IEEE". También se pueden obtener a pedido del IEEE o consultar en <http://standards.ieee.org/IPR/disclaimers.html>.

1. Descripción general

1.1 Ámbito de aplicación

Esta norma describe los diseños de software y establece el contenido de información y la organización de una descripción de diseño de software (SDD). Una SDD es una representación de un diseño de software que se utiliza para registrar información de diseño y comunicar dicha información a las partes interesadas clave del diseño. Esta norma está destinada a utilizarse en situaciones de diseño en las que se debe preparar una SDD explícita. Estas situaciones incluyen actividades de construcción de software tradicionales, cuando el diseño conduce al código, y situaciones de "ingeniería inversa" cuando se recupera una descripción de diseño de una implementación existente.

Esta norma se puede aplicar a software comercial, científico o militar que se ejecuta en computadoras digitales. La aplicabilidad no está restringida por el tamaño, la complejidad o la criticidad del software. Esta norma se puede aplicar a la descripción de diseños detallados y de alto nivel.

Esta norma no prescribe metodologías específicas para el diseño, la gestión de la configuración o el control de calidad. Esta norma no exige el uso de ningún lenguaje de diseño en particular, pero establece requisitos para la selección de lenguajes de diseño para su uso en un SDD. Esta norma se puede aplicar a la preparación de SDD capturados como documentos en papel, bases de datos automatizadas, herramientas de desarrollo de software u otros medios.

1.2 Propósito

Esta norma especifica los requisitos sobre el contenido de la información y la organización de los SDD. La norma especifica los requisitos para la selección de los lenguajes de diseño que se utilizarán para los SDD y los requisitos para documentar los puntos de vista de diseño que se utilizarán para organizar un SDD.

1.3 Público al que va dirigido

Esta norma está destinada a las partes interesadas técnicas y de gestión que preparan y utilizan descripciones de diseño. Orienta al diseñador en la selección, organización y presentación de la información de diseño. Para una organización que desarrolla sus propias prácticas de descripciones de diseño, el uso de esta norma puede ayudar a garantizar que las descripciones de diseño sean completas, concisas, coherentes, intercambiables, apropiadas para registrar las experiencias de diseño y las lecciones aprendidas, bien organizadas y fáciles de comunicar.

1.4 Conformidad

Un SDD cumple con esta norma si satisface todos los requisitos de las cláusulas 4 y 5 de esta norma. Los requisitos se indican con el verbo "deberá".

2. Definiciones

A los efectos de esta norma, se aplican los siguientes términos y definiciones. Se debe hacer referencia al Diccionario autorizado de términos de normas IEEE [B13]3 y a la norma IEEE Std 12207™-2008 [B21] para los términos no definidos en esta cláusula.

3.1 atributo de diseño: Elemento de una vista de diseño que nombra una característica o propiedad de una entidad de diseño, una relación de diseño o una restricción de diseño. Véase también: restricción de diseño, entidad de diseño, relación de diseño.

3.2 preocupación de diseño: Un área de interés con respecto al diseño de un software.

3.3 restricción de diseño: elemento de una vista de diseño que nombra y especifica una regla o restricción sobre una entidad de diseño, un atributo de diseño o una relación de diseño. Véase también: atributo de diseño, entidad de diseño, relación de diseño.

3.4 elemento de diseño: Un elemento que aparece en una vista de diseño y que puede ser cualquiera de los siguientes: entidad de diseño, relación de diseño, atributo de diseño o restricción de diseño.

3.5 entidad de diseño: Un elemento de una vista de diseño que es estructural, funcional o de otro modo distinto de otros elementos, o que desempeña un papel diferente en relación con otras entidades de diseño. Véase también: vista de diseño.

3.6 superposición de diseño: una representación de información de diseño adicional, detallada o derivada organizada con referencia a una vista de diseño existente.

3.7 fundamento del diseño: información que captura el razonamiento del diseñador que llevó al sistema tal como fue diseñado, incluidas las opciones de diseño, las compensaciones consideradas, las decisiones tomadas y las justificaciones de esas decisiones.

³ Los números entre paréntesis corresponden a los de la Bibliografía del Anexo A.

3.8 relación de diseño: Elemento de una vista de diseño que nombra una conexión o correspondencia entre entidades de diseño. Véase también: entidad de diseño.

3.9 parte interesada en el diseño: un individuo, una organización o un grupo (o clases de ellos que desempeñan el mismo papel) que tiene un interés o preocupaciones de diseño en relación con el diseño de algún elemento de software. Véase también: diseño inquietud.

3.10 sujeto de diseño: Un elemento o sistema de software para el cual se preparará un SDD. Sinónimos: software en diseño, sistema en diseño.

3.11 diseñador: La parte interesada responsable de idear y documentar el diseño del software.

3.12 vista de diseño: Representación compuesta por uno o más elementos de diseño para abordar un conjunto de cuestiones de diseño desde un punto de vista de diseño específico. Véase también: cuestión de diseño, elemento de diseño, punto de vista de diseño.

3.13 punto de vista de diseño: La especificación de los elementos y convenciones disponibles para construir y utilizar una vista de diseño. Véase también: vista de diseño.

3.14 diagrama (tipo): Un fragmento lógicamente coherente de una vista de diseño, que utiliza íconos gráficos seleccionados y convenciones para la representación visual de un lenguaje de diseño asociado, que se utilizará para representar elementos de diseño seleccionados de interés para un sistema en diseño desde un único punto de vista. Véase también: tema de diseño.

3. Modelo conceptual para descripciones de diseño de software

Esta cláusula establece un modelo conceptual para los SDD. El modelo conceptual incluye los términos y conceptos básicos de los SDD, el contexto en el que se preparan y utilizan, las partes interesadas que los utilizan y cómo se utilizan.

3.1 Diseño de software en contexto

Un diseño es una conceptualización de un tema de diseño (el sistema en diseño o el software en diseño) que incorpora sus características esenciales; demuestra un medio para cumplir con sus requisitos; sirve como base para el análisis y la evaluación y puede utilizarse para guiar su implementación. (Véase Abran y Moore [B1] para más información).

NOTA 1—Esta norma no establece qué puede ser un objeto de diseño. Un objeto de diseño puede ser cualquier elemento de software que se vaya a construir o que ya exista y se vaya a analizar. Entre los ejemplos de posibles objetos de diseño se incluyen, entre otros, sistemas, subsistemas, aplicaciones, componentes, bibliotecas, marcos de aplicación, interfaces de programación de aplicaciones (API) y catálogos de patrones de diseño⁴.

Un SDD es un producto de trabajo que describe un tema de diseño de interés. Un SDD puede producirse para capturar uno o más niveles de preocupación con respecto a su tema de diseño. Estos niveles suelen estar determinados por los métodos de diseño en uso o el contexto del ciclo de vida; tienen nombres como "diseño arquitectónico", "diseño lógico" o "diseño físico".

Un SDD se puede preparar y utilizar en una variedad de situaciones de diseño. Normalmente, un SDD se prepara para respaldar el desarrollo de un elemento de software para resolver un problema, donde este problema se ha expresado en términos de un conjunto de requisitos. El contenido del SDD se puede rastrear hasta estos requisitos. En otras palabras,

⁴ Las notas en el texto, las tablas y las figuras de una norma se proporcionan sólo con fines informativos y no contienen los requisitos necesarios para implementar esta norma.

En algunos casos, se prepara un SDD para comprender un sistema existente que carece de documentación de diseño. En tales casos, se prepara un SDD de manera que se capture, organice, presente y difunda información de interés a todas las partes interesadas. Esta información de interés se puede utilizar para la planificación, el análisis, la implementación y la evolución del sistema de software, mediante la identificación y el tratamiento de las preocupaciones esenciales de diseño.

NOTA 2—El término genérico descripción del diseño de software se utiliza en esta norma (1) para mantener la compatibilidad con la terminología de su predecesora, IEEE Std 1016-1998, y (2) para referirse a una gama de productos de trabajo que normalmente se definen en uso. Por ejemplo, la descripción del diseño de software cubre los siguientes elementos de información identificados en ISO/IEC 15289:2006 [B25]:5 descripción del diseño de la base de datos (10.14), descripción detallada del diseño de la base de datos (10.15), descripción de alto nivel del diseño de software (10.22), descripción de la interfaz (10.27), descripción de bajo nivel del diseño de software (10.29), descripción del sistema (10.71) y descripción de los elementos del sistema (10.72).

Una preocupación de diseño designa cualquier área de interés en el diseño, relacionada con su desarrollo, implementación u operación. Las preocupaciones de diseño son expresadas por las partes interesadas en el diseño. Con frecuencia, las preocupaciones de diseño surgen de requisitos específicos del software, otras surgen de restricciones contextuales. Las preocupaciones de diseño típicas incluyen funcionalidad, confiabilidad, rendimiento y capacidad de mantenimiento. Las partes interesadas típicas del diseño incluyen usuarios, desarrolladores, diseñadores de software, integradores de sistemas, mantenedores, compradores y gerentes de proyectos.

NOTA 3—Desde un punto de vista teórico de sistemas, un SDD captura el contenido de información del espacio de diseño con entradas convenientes (diagramas de diseño y especificaciones producidas por diseñadores) y salidas (resultados de transformaciones producidas por herramientas de software). El espacio de diseño típicamente contiene diseños alternativos y fundamentos de diseño además de la información mínima de la versión actual del diseño. Una propiedad interesante de una descripción de diseño como sistema es que su configuración está sujeta a evolución dinámica y el espacio de estado respectivo, basado en sus elementos de diseño, no se da de antemano sino que se crea iterativamente a modo de análisis de sistemas por síntesis. La síntesis de diseño final se obtiene a través de análisis sucesivos de diseños intermedios. Por lo tanto, un SDD puede considerarse un sistema abierto, dirigido a objetivos, cuyo estado final es un modelo detallado del sistema en diseño.

Un SDD se organiza mediante vistas de diseño. Una vista de diseño aborda una o más de las cuestiones de diseño.

NOTA 4—El uso de múltiples vistas para lograr la separación de intereses tiene una larga historia en la ingeniería de software (ver Ross, Goodenough e Irvine [B33] y Ross [B31]), recientemente en la ingeniería de requisitos orientada a puntos de vista (ver Nuseibeh, Kramer y Finkelstein [B27]), y particularmente relevante para esta norma, el uso de vistas para presentar racionalmente los resultados de un proceso de diseño (ver Parnas y Clements [B30]) y su uso durante el diseño (ver Gomma [B11]). La formulación particular aquí se deriva de IEEE Std 1471™-2000 [B20].

Cada vista de diseño está regida por un punto de vista de diseño. Cada punto de vista de diseño se centra en un conjunto de cuestiones de diseño e introduce un conjunto de recursos descriptivos denominados elementos de diseño que se utilizan para construir e interpretar la vista de diseño.

Ejemplo:

Un punto de vista puede introducir elementos de diseño familiares, como funciones, entradas y salidas; estos elementos se utilizan para construir una vista funcional.

Existen cuatro tipos de elementos de diseño: entidades de diseño, relaciones de diseño, atributos de diseño y restricciones de diseño. Un punto de vista de diseño determina los tipos de elementos de diseño que se utilizarán en cualquier vista de diseño que controle. Cada vista de diseño se expresa como una colección de instancias de entidades de diseño, atributos de diseño, relaciones de diseño entre entidades de diseño y restricciones de diseño sobre esos elementos. Las necesidades de información de diseño de las partes interesadas del sistema en diseño se deben satisfacer mediante el uso de estos elementos.

NOTA 5—Aunque una vista no necesita ser un gráfico, su contenido se describe frecuentemente utilizando diagramas que pueden formalizarse como extensiones o especializaciones de gráficos conceptuales de elementos de diseño (véase el borrador de Gráficos Conceptuales de ISO). [B22]).

⁵ Las expresiones entre paréntesis en el resto de esta NOTA se refieren a las subcláusulas de ISO/IEC 15289:2006 [B25], donde estas se definen elementos de información.

A veces resulta útil recopilar y presentar información que no siga estrictamente la división de la información según los puntos de vista del diseño. Una superposición de diseño es un mecanismo para organizar y presentar dicha información adicional sobre el diseño.

El diseño implica la consideración y evaluación de alternativas y compensaciones entre alternativas que conducen a decisiones (véase Abran y Moore [B1]). Las decisiones de diseño y la justificación del diseño para las decisiones se capturan tanto para ayudar a la comprensión de las partes interesadas actuales del diseño como para respaldar la toma de decisiones futuras (véase IEEE Std 12207-2008 [B21]).

Los conceptos clave del SDD se representan en la Figura 1a y la Figura 1b.

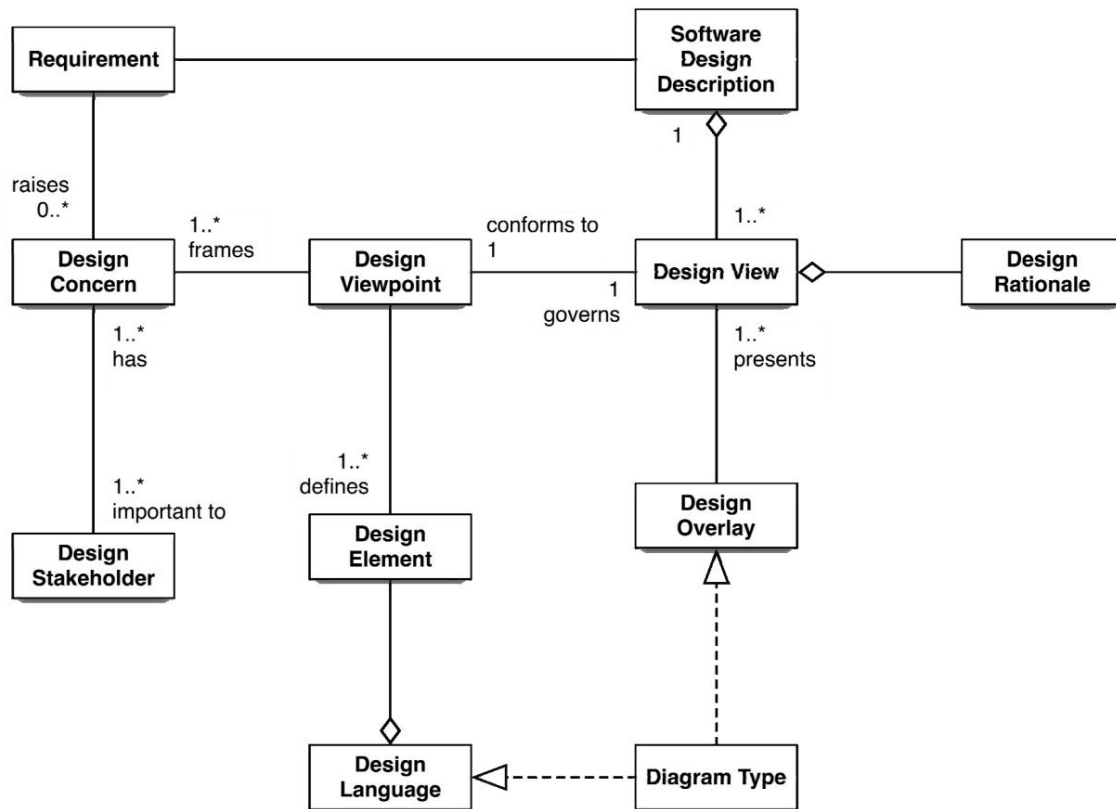


Figura 1 a—Modelo conceptual: vista superior

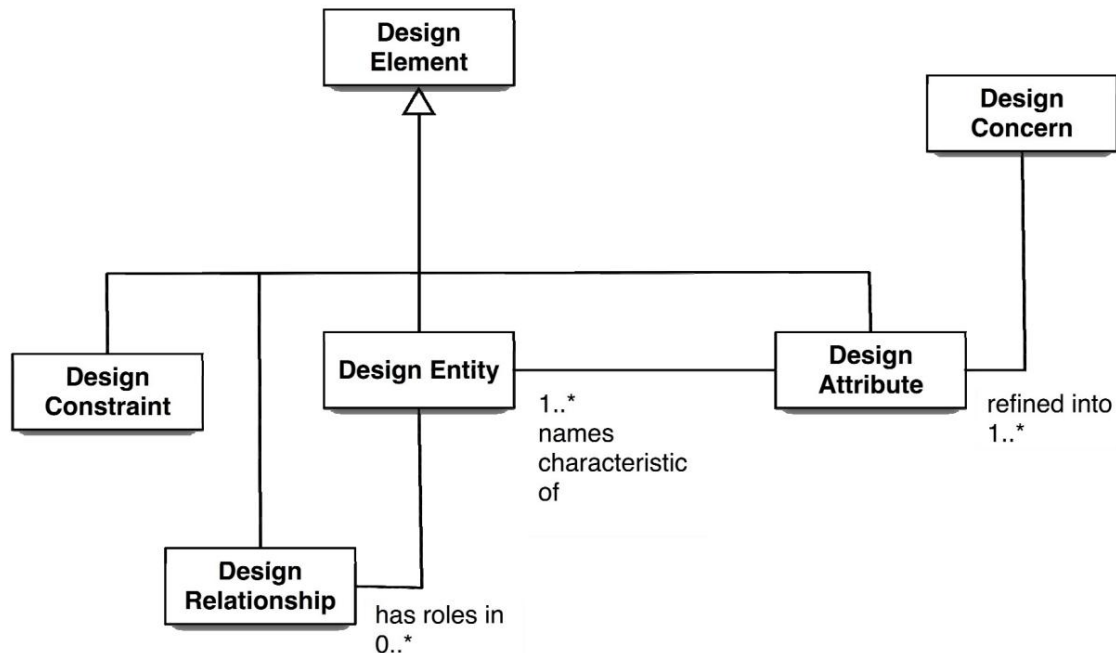


Figura 1b—Modelo conceptual: elementos de diseño

NOTA 6—La Figura 1a y la Figura 1b proporcionan un resumen de los conceptos clave utilizados en esta norma y sus relaciones. La figura presenta estos conceptos y sus relaciones desde la perspectiva de un único SDD utilizado para representar un único tema de diseño. Un SDD incluye una o más vistas de diseño que emplean puntos de vista de diseño seleccionados para cubrir las preocupaciones de diseño de las partes interesadas planteadas por los requisitos. El contenido de cada vista de diseño se expresa en términos de entidades de diseño, sus atributos de diseño y relaciones de diseño, utilizando lenguajes de diseño seleccionados. En la figura, los cuadros representan clases de cosas. Las líneas que conectan los cuadros representan asociaciones entre clases de cosas. Cada clase que participa en una asociación tiene un rol en esa asociación. Un rol se nombra opcionalmente con una etiqueta que aparece cerca de donde la asociación está unida a la clase. Por ejemplo, en la asociación entre el punto de vista de diseño y el elemento de diseño, el rol del elemento de diseño se etiqueta como define. Cada rol puede tener una multiplicidad, que denota un número o un conjunto de números como un rango numérico. Un diamante al final de una asociación denota una relación de parte de. Por ejemplo, la asociación entre la vista de diseño y la descripción del diseño de software se interpreta como "una o más vistas de diseño son parte de una descripción del diseño de software". Esta notación se define en OMG formal/2007-11-2 y 2007-11-4, Lenguaje de modelado unificado [B28], [B29].

3.2 Descripciones del diseño de software dentro del ciclo de vida

En esta norma, se utilizará un ciclo típico para describir las distintas situaciones de diseño en las que se puede crear y utilizar un SDD. Este ciclo de vida se basa en la norma IEEE Std 12207-2008 [B21].

3.2.1 Influencias en la preparación de SDD

El producto clave del ciclo de vida del software que impulsa el diseño de un software es, por lo general, la especificación de requisitos del software (SRS). Una SRS captura los requisitos del software que impulsarán el diseño y las restricciones de diseño que se deben considerar o respetar (consulte la norma IEEE Std 830™-1998 [B15] e ISO/IEC 15289:2006 [B25]).

3.2.2 Influencias en el ciclo de vida del software de los productos

El SDD influye en el contenido de varios productos de trabajo importantes del ciclo de vida del software. Los desarrolladores de estos productos serán reconocidos entre el público al que va dirigido el SDD.

Especificación de requisitos de software. Las decisiones de diseño o las limitaciones de diseño descubiertas durante la preparación de la especificación de requisitos de software pueden dar lugar a cambios en los requisitos. Mantener la trazabilidad entre los requisitos y el diseño es una forma de registrar la relación entre ambos (consulte Abran y Moore [B1]).

Documentación de pruebas. Los planes de pruebas pueden verse influenciados por el SDD. El desarrollo de casos de prueba y Los procedimientos de prueba pueden tener en cuenta el contenido del SDD.

3.2.3 Verificación del diseño y función del diseño en la validación

La verificación es la determinación de si un producto de software cumple con los requisitos especificados (consulte la norma IEEE Std 12207-2008 [B21]). Un SDD está sujeto a la verificación del diseño para determinar si el diseño: aborda las preocupaciones de diseño de las partes interesadas; es coherente con los requisitos establecidos; implementa las decisiones de diseño previstas (como las relacionadas con las interfaces, las entradas, las salidas, los algoritmos, la asignación de recursos y el manejo de errores); logra las cualidades previstas (como la seguridad, la protección o la capacidad de mantenimiento); y se ajusta a una arquitectura impuesta. Por lo tanto, la verificación puede introducir preocupaciones de diseño que se abordarán en un SDD.

Su verificación de diseño generalmente se realizará mediante revisión, inspección o análisis (consulte IEEE Std 1028™-2008 [B17]).

La validación es la determinación de que se cumplen los requisitos para un uso específico previsto de un producto de software (véase la norma IEEE Std 12207-2008 [B21]). El SDD desempeña un papel en la validación al proporcionar: una descripción general necesaria para comprender la implementación; la justificación de las decisiones de diseño tomadas; y la trazabilidad hasta los requisitos del software en diseño. (Véase la norma IEEE Std 1012™-2004 [B16] sobre verificación y validación).

4. Contenido de la información de descripción del diseño

4.1 Introducción

Los contenidos requeridos de un SDD son los siguientes:

Identificación del SDD

Partes interesadas en el diseño identificadas

Preocupaciones de diseño identificadas

Puntos de vista de diseño seleccionados, cada uno con definiciones de tipo de sus elementos de diseño permitidos y diseño idiomas

Vistas de diseño

Superposiciones de diseño

Fundamento del diseño

Estos se describen en el resto de esta cláusula.

4.2 Identificación de SDD

Una SDD deberá incluir la siguiente información descriptiva:

Fecha de emisión y estado

Alcance

Organismo emisor

Autoría (información sobre responsabilidad o derechos de autor)

Referencias

Context

Uno o más lenguajes de diseño para cada punto de vista de diseño utilizado

Cuerpo

Resumen

Glosario

Historial de cambios

NOTA: Este requisito permite que un SDD cumpla con las pautas sobre una descripción en 7.3 de ISO/IEC 15289:2006 [B25]. La descripción de los lenguajes de diseño es una parte adecuada de las declaraciones del punto de vista del diseño (según 4.5). El cuerpo del SDD está organizado en vistas de diseño (según 4.4), posiblemente con superposiciones asociadas (según 4.7) y fundamentos del diseño (según 4.8).

4.3 Las partes interesadas en el diseño y sus preocupaciones

Un SDD deberá identificar a las partes interesadas en el diseño para el tema del diseño.

Un SDD deberá identificar las preocupaciones de diseño de cada parte interesada en el diseño identificada.

Una SDD deberá abordar cada problema de diseño identificado.

NOTA: Se puede utilizar una SDD para satisfacer las directrices de contenido para varios tipos de descripción de diseño, tal como se define en la norma ISO/IEC 15289:2006 [B25], identificando sus directrices de contenido como cuestiones de diseño. Los tipos de descripciones de diseño son los siguientes: descripción de diseño de base de datos (10.14),⁶ descripción detallada de diseño de base de datos (10.15), descripción de diseño de software de alto nivel (10.22), descripción de interfaz (10.27), descripción de diseño de software de bajo nivel (10.29), descripción de sistema (10.71) y descripción de elemento de sistema (10.72).

4.4 Vistas de diseño

Un SDD deberá organizarse en una o más vistas de diseño.

Cada vista de diseño en un SDD deberá ajustarse a su punto de vista de diseño rector.

El propósito de una vista de diseño es abordar las preocupaciones de diseño relacionadas con el tema del diseño, para permitir que una parte interesada en el diseño se centre en los detalles del diseño desde una perspectiva específica y abordar de manera eficaz los requisitos relevantes.

⁶ Las expresiones entre paréntesis en el resto de esta NOTA se refieren a las subcláusulas de ISO/IEC 15289:2006 [B25], donde se definen las pautas de contenido para cada elemento de información.

Cada vista de diseño deberá abordar las preocupaciones de diseño especificadas por el punto de vista de diseño que la rige.

Un SDD está completo cuando cada preocupación de diseño identificada es el tema de al menos una vista de diseño; todos los atributos de diseño refinados a partir de cada preocupación de diseño desde algún punto de vista se han especificado para todas las entidades y relaciones de diseño en su vista asociada; y se han aplicado todas las restricciones de diseño.

Un SDD es consistente si no hay conflictos conocidos entre los elementos de diseño de sus vistas de diseño.

NOTA: Los usuarios de esta norma pueden desear indicar los requisitos de entrega en un SDD en términos de las nociones anteriores de integridad y consistencia.

4.5 Puntos de vista del diseño

Para cada vista de diseño en un SDD, deberá haber exactamente un punto de vista de diseño que la gobierne.

Cada punto de vista del diseño deberá especificarse mediante:

Nombre del punto de vista;

Preocupaciones de diseño que son los temas del punto de vista;

Elementos de diseño, definidos por ese punto de vista, específicamente los tipos de entidades de diseño, atributos, relaciones y restricciones introducidas por ese punto de vista o utilizadas por ese punto de vista (que pueden haberse definido en otra parte). Estos elementos pueden implementarse mediante uno o más lenguajes de diseño;

Métodos analíticos u otras operaciones que se utilizarán para construir una vista de diseño basada en la punto de vista y criterios para interpretar y evaluar el diseño; y

Fuente del punto de vista (por ejemplo, autoría o cita), cuando corresponda.

Además, una especificación del punto de vista del diseño puede proporcionar la siguiente información sobre el uso del punto de vista:

Pruebas formales o informales de consistencia y completitud que se aplicarán a la vista;

Técnicas de evaluación o análisis que se aplicarán a una vista; y

Heurísticas, patrones u otras pautas para ayudar en la construcción o síntesis de una vista.

Una SDD deberá incluir una justificación para la selección de cada punto de vista seleccionado.

Cada problema de diseño identificado en una descripción detallada de la estructura (4.3) deberá estar enmarcado en al menos un punto de vista de diseño seleccionado para su uso. Un problema de diseño puede ser el foco de más de un punto de vista en una descripción detallada de la estructura.

NOTA: Se puede incluir una especificación del punto de vista del diseño en la SDD o incorporarla por referencia.

4.6 Elementos de diseño

Un elemento de diseño es cualquier elemento que aparece en una vista de diseño. Un elemento de diseño puede ser cualquiera de los siguientes subcasos: entidad de diseño, relación de diseño, atributo de diseño o restricción de diseño.

Cada elemento de diseño en el SDD deberá tener un nombre (4.6.2.1), un tipo (4.6.2.2) y cualquier contenido.

NOTA 1—Este requisito es “heredado” por los cuatro subcasos: entidades de diseño, relaciones de diseño, atributos de diseño y restricciones de diseño.

El tipo de cada elemento de diseño deberá introducirse dentro de exactamente una definición de punto de vista de diseño.

Un elemento de diseño se puede utilizar en una o más vistas de diseño.

NOTA 2—Un elemento de diseño es introducido y “poseído” por exactamente una vista de diseño, de acuerdo con su definición de tipo dentro del punto de vista asociado. Puede ser compartido o referenciado dentro de otras vistas de diseño. Compartir elementos de diseño permite la expresión de aspectos de diseño como en el diseño orientado a aspectos.

4.6.1 Entidades de diseño

Las entidades de diseño capturan elementos clave de un diseño de software.

Cada entidad de diseño tendrá un nombre (4.6.2.1), un tipo (4.6.2.2) y un propósito (4.6.2.3).

Los ejemplos de entidades de diseño incluyen, entre otros, los siguientes: sistemas, subsistemas, bibliotecas, marcos, patrones de colaboración abstractos, plantillas genéricas, componentes, clases, almacenes de datos, módulos, unidades de programa, programas y procesos.

NOTA: La cantidad y los tipos de entidades necesarias para expresar una vista de diseño dependen de varios factores, como la complejidad del sistema, la técnica de diseño utilizada y el entorno de soporte de la herramienta.

4.6.2 Atributos de diseño

Un atributo de diseño nombra una característica o propiedad de un elemento de diseño (que puede ser una entidad de diseño, una restricción de diseño o una relación de diseño) y proporciona una declaración de hechos sobre ese elemento de diseño.

Se deberán especificar todos los atributos de diseño declarados por un punto de vista de diseño.

NOTA 1—Los atributos de diseño pueden considerarse como preguntas sobre los elementos de diseño. Las respuestas a esas preguntas son los valores de los atributos. Se pueden responder todas las preguntas, pero el contenido de la respuesta dependerá de la naturaleza de la entidad. La colección de respuestas proporciona una descripción completa de una entidad. Las descripciones de los atributos deben incluir referencias y consideraciones de diseño, como pros y contras y suposiciones, cuando corresponda. En algunos casos, las descripciones de los atributos pueden tener el valor none.

NOTA 2—Los atributos de diseño se han generalizado a partir del concepto de atributo de entidad de diseño (que apareció en la norma IEEE Std 1016-1998 y se aplicaba solo a entidades de diseño) para aplicarse a entidades de diseño, relaciones de diseño y restricciones de diseño.

NOTA 3—El uso de los atributos de diseño de 4.6.2.1 a 4.6.2.3 garantiza la compatibilidad con la norma IEEE Std 1016-1998. Otros atributos de diseño requeridos como parte de puntos de vista de diseño específicos se definen con esos puntos de vista en la Cláusula 5. Algunos atributos de diseño [como los subordinados (véase 5.3.2.2)] se pueden representar de manera más útil como relaciones de diseño. Esto no era posible en la norma IEEE Std 1016-1998.

4.6.2.1 Atributo de nombre

El nombre del elemento. Cada elemento de diseño deberá tener un nombre de referencia inequívoco. Los nombres de los elementos pueden seleccionarse para caracterizar su naturaleza. Esto simplificará la referencia y el seguimiento, además de proporcionar identificación.

4.6.2.2 Atributo de tipo

Una descripción del tipo de elemento. El atributo de tipo debe describir la naturaleza del elemento. Puede simplemente nombrar el tipo de elemento, como subsistema, componente, marco, biblioteca, clase, subprograma, etc.

módulo, unidad de programa, función, procedimiento, proceso, objeto, objeto persistente, clase o almacén de datos.

Alternativamente, los elementos de diseño pueden agruparse en clases principales para ayudar a localizar un elemento que trate un tipo particular de información.

Dentro de un SDD, los tipos de elementos elegidos se deben aplicar de forma consistente.

4.6.2.3 Atributo de propósito

Una descripción de por qué existe el elemento. El atributo de propósito debe proporcionar la justificación para la creación del elemento.

4.6.2.4 Atributo de autor

Identificación del diseñador. El atributo autor debe identificar al diseñador del elemento.

4.6.3 Relaciones de diseño

Una relación de diseño designa una asociación o correspondencia entre dos o más entidades de diseño. Proporciona una declaración de hechos sobre esas entidades de diseño.

Cada relación de diseño en un SDD debe tener un nombre (4.6.2.1) y un tipo (4.6.2.2). Una relación de diseño debe identificar las entidades de diseño que participan en la relación.

NOTA: En esta norma no se definen relaciones de diseño. La mayoría de las técnicas de diseño utilizan relaciones de diseño de forma extensiva. Normalmente, estas relaciones de diseño se definirán como parte de un punto de vista de diseño. Por ejemplo, los métodos de diseño estructurado se construyen en torno a relaciones de diseño que incluyen relaciones de entrada (el dato I es una entrada para el proceso A), de salida (el dato O es una salida del proceso A) y de descomposición (el proceso A se descompone en los procesos A1, A2 y A3). Los métodos de diseño orientados a objetos utilizan relaciones de diseño que incluyen encapsulación, generalización, especialización, composición, agregación, realización e instanciación.

4.6.4 Restricciones de diseño

Una restricción de diseño es un elemento de una vista de diseño que nombra una regla o restricción impuesta por un elemento de diseño (la fuente) sobre otro elemento de diseño (el destino), que puede ser una entidad de diseño, un atributo de diseño o una relación de diseño.

Cada restricción de diseño en un SDD debe tener un nombre (4.6.2.1) y un tipo (4.6.2.2). Una restricción de diseño debe identificar sus entidades de diseño de origen y destino.

NOTA: No existen restricciones de diseño predefinidas en esta norma. Muchas técnicas de diseño introducen restricciones de diseño.

4.7 Superposiciones de diseño

Una superposición de diseño se utiliza para presentar información adicional con respecto a una vista de diseño ya definida.

Cada superposición de diseño deberá tener un nombre único y estar marcada como superposición.

Cada superposición de diseño deberá estar claramente asociada a un único punto de vista.

NOTA: Las razones para utilizar una superposición de diseño como parte de un SDD incluyen: proporcionar un mecanismo de extensión para que la información de diseño se presente de manera conveniente sobre alguna vista sin el requisito de una estandarización externa existente de lenguajes y notaciones para dicha representación; extender el poder expresivo de la representación con detalles adicionales mientras se reutiliza información de vistas existentes (es decir, sin la necesidad de definir vistas adicionales o almacenar de manera persistente información de diseño derivable); y relacionar la información de diseño con hechos del entorno del sistema para la conveniencia del diseñador (u otras partes interesadas).

4.8 Fundamento del diseño

La justificación del diseño captura el razonamiento del diseñador que llevó al sistema tal como fue diseñado y las justificaciones de esas decisiones.

La justificación del diseño puede adoptar la forma de comentarios, realizados a lo largo del proceso de decisión y asociados con conjuntos de elementos de diseño. La justificación del diseño puede incluir, entre otros: cuestiones de diseño planteadas y abordadas en respuesta a inquietudes de diseño; opciones de diseño consideradas; compensaciones evaluadas; decisiones tomadas; criterios utilizados para guiar las decisiones de diseño; y argumentos y justificaciones presentados para tomar decisiones.

NOTA: La única justificación de diseño requerida es el uso del atributo de propósito (4.6.2.3).

4.9 Lenguajes de diseño

Los lenguajes de diseño se seleccionan como parte de la especificación del punto de vista del diseño (4.5).

Se puede seleccionar un lenguaje de diseño para un punto de vista de diseño solo si admite todos los elementos de diseño definidos por ese punto de vista.

Se seleccionarán lenguajes de diseño que tengan:

- Una sintaxis y una semántica bien definidas; y

- El estado de una norma disponible o un documento definitorio equivalente.

En un SDD solo se deben utilizar lenguajes de diseño estandarizados y bien establecidos (es decir, definidos previamente y fácilmente disponibles). En el caso de un lenguaje de diseño recién inventado, la definición del lenguaje debe proporcionarse como parte de la declaración del punto de vista.

NOTA 1—Los lenguajes de diseño estandarizados que se usan comúnmente son preferibles a los lenguajes establecidos sin una definición formal. Algunos ejemplos de lenguajes estandarizados incluyen: IDEF0 (IEEE Std 1320.1™-1998 [B18]); IDEF1X (IEEE Std 1320.2™-1998 [B19]); Lenguaje de modelado unificado (UML) (OMG [B28] y [B29]); Método de definición de Viena (VDM) (ISO/IEC 13817-1:1996 [B24]); y Z (ISO/IEC 13568:2002 [B23]). Algunos ejemplos de lenguajes establecidos incluyen: máquinas de estados, autómatas, tablas de decisión, diagramas de Warnier, Jackson Structured Design (JSD), lenguajes de diseño de programas (PDL), diagramas de estructura, Jerarquía más entrada-proceso-salida (HIPO), modelos de confiabilidad y modelos de colas.

NOTA 2: Es aceptable utilizar un lenguaje de diseño en más de una vista. También es aceptable utilizar más de un lenguaje de diseño dentro de cualquier número de vistas cuando cada lenguaje de diseño que se utilizará es declarado por el punto de vista. Esto es aceptable incluso para una parte del diseño; por ejemplo, cuando se utiliza como base para el intercambio; debido a consideraciones organizativas como el desarrollo por parte de miembros del equipo que no comparten el mismo puesto; subcontratación de la responsabilidad parcial del diseño; o aprovechamiento de herramientas de diseño particulares o de la experiencia del diseñador.

NOTA 3—El Anexo B establece un formato uniforme para describir los lenguajes de diseño que se utilizarán en los SDD.

NOTA 4—En caso de que no haya disponible un lenguaje de diseño adecuado para un punto de vista específico, es responsabilidad del diseñador proporcionar una adaptación de un lenguaje existente o la definición de un nuevo lenguaje de diseño apropiado.

El diseñador proporcionaría esta definición del lenguaje de diseño para incluirla en el SDD de acuerdo con los requisitos para los puntos de vista en 4.5.

5. Puntos de vista del diseño

5.1 Introducción

Esta cláusula define varios puntos de vista de diseño para su uso en SDD. Ilustra la realización de estos puntos de vista de diseño en términos de selecciones de lenguaje de diseño, relaciona las preocupaciones de diseño con los puntos de vista y establece nombres neutrales en cuanto al lenguaje (notación y método) para estos puntos de vista. La Tabla 1 resume estos puntos de vista de diseño. Para cada punto de vista, se enumeran su nombre, preocupaciones de diseño y lenguajes de diseño apropiados. Se proporcionan breves descripciones relacionadas con un conjunto mínimo de entidades de diseño, relaciones de diseño, atributos de entidades de diseño y restricciones de diseño para cada punto de vista. También se incluyen referencias adicionales pertinentes al uso de cada punto de vista.

El punto de vista de diseño definido en esta cláusula se utilizará en la SDD siempre que sea aplicable al tema del diseño, en función de las preocupaciones de diseño identificadas (4.3).

Tabla 1—Resumen de los puntos de vista del diseño

Punto de vista del diseño	Preocupaciones de diseño	Ejemplos de lenguajes de diseño
Contexto (5.2)	Servicios y usuarios del sistema	IDEF0, diagrama de casos de uso UML, Diagrama de contexto de análisis estructurado
Composición (5.3) Se puede refinar en nuevos puntos de vista, como: descomposición funcional (lógica) y descomposición en tiempo de ejecución (física).	Composición y ensamblaje modular de sistemas en términos de subsistemas y componentes (enchufables), comprar vs. construir, reutilización de componentes	Lógico: Diagrama de paquetes UML, Diagrama de componentes UML, Lenguajes de descripción de arquitectura, IDEF0, Diagrama de estructura, HIPO Físico: Diagrama de implementación UML
Lógico (5.4)	Estructura estática (clases, interfaces y sus relaciones) Reutilización de tipos e implementaciones (clases, tipos de datos)	Diagrama de clases UML, diagrama de objetos UML
Dependencia (5.5)	Interconexión, compartición y parametrización	Diagrama de paquetes UML y diagrama de componentes
Información (5.6) con superposición de distribución de datos y superposición volumétrica física Patrones	Información persistente	IDEF1X, diagrama entidad-relación, Diagrama de clases UML
(5.7)	Reutilización de patrones y recursos disponibles Plantilla de marco	Diagrama de estructura compuesta UML
Interfaz (5.8)	Definición del servicio, acceso al servicio	Lenguajes de definición de interfaz (IDL), Diagrama de componentes UML
Estructura (5.9)	Constituyentes internos y organización de los sujetos, componentes y clases de diseño	Diagrama de estructura UML, diagrama de clases
Interacción (5.10)	Comunicación de objetos, mensajería	Diagrama de secuencia UML, diagrama de comunicación UML
Dinámica de estados (5.11)	Transformación de estado dinámica	Diagrama de máquina de estados UML, diagrama de estados (de Harel), tabla de transición de estados (matriz), autómatas, red de Petri
Algoritmo (5.12)	Lógica procedimental	Tabla de decisiones, diagrama de Warnier, JSP, PDL
Recursos (5.13) Puede refinarse en puntos de vista basados en recursos con posibles superposiciones.	Utilización de recursos	Perfil en tiempo real UML, diagrama de clases UML, restricción de objetos UML Lenguaje (OCL)

5.2 Punto de vista del contexto

El punto de vista del contexto describe los servicios proporcionados por un sujeto de diseño con referencia a un contexto explícito. Ese contexto se define por referencia a actores que incluyen usuarios y otras partes interesadas, que interactúan con el sujeto de diseño en su entorno. El punto de vista del contexto proporciona una perspectiva de “caja negra” sobre el sujeto de diseño.

Los servicios representan un aspecto inherentemente funcional o casos previstos de uso del sujeto de diseño (de ahí el término “casos de uso” en UML). La estratificación de los servicios y sus descripciones en forma de escenarios de interacciones de los actores con el sistema proporcionan un mecanismo para añadir detalles. Los servicios también pueden asociarse con los actores a través de flujos de información. El contenido y la forma de intercambio de información con el entorno implican información de diseño adicional y la necesidad de puntos de vista adicionales (véase 5.10).

Una superposición de implementación en una vista de contexto se puede transformar en una vista de implementación siempre que la plataforma de hardware de ejecución sea parte del tema de diseño; para el diseño de software independiente, una superposición de implementación asigna entidades de software a entidades disponibles externamente que no son tema del esfuerzo de diseño actual. De manera similar, la asignación de trabajo a los equipos y otras perspectivas de gestión son superposiciones en el diseño.

5.2.1 Preocupaciones de diseño

El propósito del punto de vista del Contexto es identificar los servicios ofrecidos por un sujeto de diseño, sus actores (usuarios y otras partes interesadas que interactúan), establecer el límite del sistema y delinear efectivamente el alcance de uso y operación del sujeto de diseño.

Trazar un límite que separe al sujeto de diseño de su entorno, determinar un conjunto de servicios que se proporcionarán y los flujos de información entre el sujeto de diseño y su entorno es, por lo general, una decisión clave de diseño. Eso hace que este punto de vista sea aplicable a la mayoría de los esfuerzos de diseño.

Cuando el sistema se presenta como una caja negra, con decisiones internas ocultas, la vista de contexto es a menudo un punto de partida del diseño, mostrando lo que se debe diseñar funcionalmente como la única información disponible sobre el sujeto de diseño: un nombre y un conjunto asociado de servicios identificables externamente. El análisis de requisitos identifica estos servicios con la especificación de atributos de calidad del servicio, invocando de ahí en adelante muchos requisitos no funcionales. Con frecuencia incompleta, se comienza una vista de contexto en el análisis de requisitos. El trabajo para completar esta vista continúa durante el diseño.

5.2.2 Elementos de diseño

Entidades de diseño: actores (elementos activos externos que interactúan con el sujeto de diseño, incluidos usuarios, otras partes interesadas y sistemas externos u otros elementos); servicios (también llamados casos de uso); y flujos de información dirigidos entre el sujeto de diseño, tratado como una caja negra, y sus actores que asocian actores con servicios. Los flujos capturan el contenido de información esperado intercambiado.

Relaciones de diseño: recibir resultados y proporcionar información (entre los actores y el sujeto del diseño).

Restricciones de diseño: cualidades del servicio; forma y medio de interacción (proporcionada y recibida) con el entorno.

5.2.3 Ejemplos de idiomas

Se puede utilizar cualquier diagrama de tipo caja negra para realizar el punto de vista del contexto. Los lenguajes apropiados incluyen análisis estructurado [p. ej., IDEF0 (IEEE Std 1320.1-1998 [B18]), análisis estructurado y diseño

Técnica (SADT) (Ross [B32]) o los de la variedad DeMarco o Gane-Sarson], los diagramas de caja negra de la sala limpia y los casos de uso de UML (OMG [B28]).

5.3 Punto de vista de la composición

El punto de vista de la composición describe la forma en que el tema de diseño se estructura (recursivamente) en partes constituyentes y establece los roles de esas partes.

5.3.1 Preocupaciones de diseño

Los desarrolladores y mantenedores de software utilizan este punto de vista para identificar los principales componentes de diseño del tema de diseño, para localizar y asignar funcionalidad, responsabilidades u otros roles de diseño a estos componentes. En el mantenimiento, se puede utilizar para realizar análisis de impacto y localizar los esfuerzos de realizar cambios. También se puede abordar la reutilización, a nivel de subsistemas existentes y componentes de granularidad grande. La información en una vista de Composición puede ser utilizada por la administración de adquisiciones y en la administración de proyectos para la especificación y asignación de paquetes de trabajo, y para la planificación, monitoreo y control de un proyecto de software. Esta información, junto con otra información del proyecto, se puede utilizar para estimar el costo, la dotación de personal y el cronograma para el esfuerzo de desarrollo. La administración de configuración puede utilizar la información para establecer la organización, el seguimiento y la administración de cambios de productos de trabajo emergentes (consulte IEEE Std 12207-2008 [B21]).

5.3.2 Elementos de diseño

Entidades de diseño: tipos de constituyentes de un sistema: subsistemas, componentes, módulos; puertos e interfaces (proporcionadas y requeridas); también bibliotecas, marcos, repositorios de software, catálogos y plantillas.

Relaciones de diseño: composición, uso y generalización. El punto de vista Composición permite registrar las relaciones parte-todo entre entidades de diseño mediante relaciones de realización, dependencia, agregación, composición y generalización. Se requieren y proporcionan relaciones de diseño adicionales (interfaces) y la conexión de puertos a componentes.

Atributos de diseño: para cada entidad de diseño, el punto de vista proporciona una referencia a una descripción detallada a través del atributo de identificación. Se deben utilizar las descripciones de los atributos de identificación, tipo, propósito, función y definición.

5.3.2.1 Atributo de función

Una declaración de lo que hace la entidad. El atributo de función indica la transformación que aplica la entidad a sus entradas para producir la salida. En el caso de una entidad de datos, este atributo indica el tipo de información almacenada o transmitida por la entidad.

NOTA: Este atributo de diseño se conserva por compatibilidad con la norma IEEE Std 1016-1998.

5.3.2.2 Atributo de subordinados

La identificación de todas las entidades que componen esta entidad. El atributo de subordinados identifica la relación de "composición de" para una entidad. Esta información se utiliza para rastrear requisitos hasta entidades de diseño y para identificar relaciones estructurales padre/hijo a través de un sujeto de diseño.

NOTA: Este atributo de diseño se mantiene por compatibilidad con la norma IEEE 1016-1998. Existe una capacidad equivalente disponible a través de la relación de composición.

5.3.3 Ejemplos de idiomas

Los diagramas de componentes UML (véase OMG [B28]) cubren este punto de vista. La técnica gráfica más simple que se utiliza para describir la descomposición funcional del sistema es un diagrama de descomposición jerárquica; dicho diagrama se puede utilizar junto con descripciones en lenguaje natural del propósito y la función de cada entidad, como las que se proporcionan en IDEF0 (IEEE Std 1320.1-1998 [B18]), el diagrama de estructura (Yourdon y Constantine [B38]) y el diagrama HIPO. La composición en tiempo de ejecución también puede utilizar diagramas estructurados (Page-Jones [B29]).

5.4 Punto de vista lógico

El objetivo del punto de vista lógico es elaborar tipos existentes y diseñados y sus implementaciones como clases e interfaces con sus relaciones estáticas estructurales. Este punto de vista también utiliza ejemplos de instancias de tipos para esbozar ideas de diseño.

5.4.1 Preocupaciones de diseño

El punto de vista lógico se utiliza para abordar el desarrollo y la reutilización de abstracciones adecuadas y sus implementaciones. Para cualquier plataforma de implementación, hay disponible un conjunto de tipos para las abstracciones de dominio de interés en un tema de diseño, y se debe diseñar una serie de tipos nuevos, algunos de los cuales pueden considerarse para su reutilización. La principal preocupación es la elección adecuada de las abstracciones y su expresión en términos de tipos existentes (algunos de los cuales pueden haber sido específicos del tema de diseño).

5.4.2 Elementos de diseño

Entidades de diseño: clase, interfaz, tipo de energía, tipo de datos, objeto, atributo, método, clase de asociación, plantilla y espacio de nombres.

Relaciones de diseño: asociación, generalización, dependencia, realización, implementación, instancia de, composición y agregación.

Atributos de diseño: nombre, nombre del rol, visibilidad, cardinalidad, tipo, estereotipo, redefinición, valor etiquetado, parámetro y eficiencia de navegación.

Restricciones de diseño: restricciones de valor, restricciones de exclusividad de relaciones, navegabilidad, conjuntos de generalización, multiplicidad, derivación, variabilidad, valor inicial, calificador, ordenamiento, estático, precondition, postcondición y restricciones de conjunto de generalización.

5.4.3 Ejemplos de idiomas

Diagramas de clases UML y diagramas de objetos UML (que muestran objetos como instancias de sus respectivas clases) (OMG [B28]). Las redes de tipos y las referencias a tipos implementados se utilizan comúnmente como información complementaria.

5.5 Punto de vista de dependencia

El punto de vista de dependencia especifica las relaciones de interconexión y acceso entre entidades. Estas relaciones incluyen información compartida, orden de ejecución o parametrización de interfaces.

5.5.1 Preocupaciones de diseño

Una vista de dependencia proporciona una imagen general del tema de diseño para evaluar el impacto de los requisitos o los cambios de diseño. Puede ayudar a los encargados del mantenimiento a aislar las entidades que causan fallas del sistema o cuellos de botella de recursos. Puede ayudar a producir el plan de integración del sistema al identificar las entidades que necesitan otras entidades y que deben desarrollarse primero. Esta descripción también puede ser utilizada por las pruebas de integración para ayudar en la producción de casos de prueba de integración.

5.5.2 Elementos de diseño

Entidades de diseño: subsistema, componente y módulo.

Relaciones de diseño: usos, proporciona y requiere.

Atributo de diseño: nombre (4.6.2.1), tipo (4.6.2.2), propósito (4.6.2.3), dependencias (5.5.2.1) y recursos.

Estos atributos deben proporcionarse para todas las entidades de diseño.

5.5.2.1 Atributo de dependencias

Descripción de las relaciones de esta entidad con otras entidades. El atributo de dependencias identifica los usos o exige la presencia de relaciones para una entidad. Este atributo se utiliza para describir la naturaleza de cada interacción, incluidas características como el tiempo y las condiciones de la interacción. Las interacciones implican la iniciación, el orden de ejecución, el uso compartido de datos, la creación, la duplicación, el uso, el almacenamiento o la destrucción de entidades.

NOTA: Este atributo de entidad de diseño se conserva por compatibilidad con la norma IEEE Std 1016-1998.

5.5.3 Ejemplos de idiomas

Diagramas de componentes UML y diagramas de paquetes UML que muestran dependencias entre subsistemas (OMG [B28]).

5.6 Punto de vista de la información

El punto de vista de la información es aplicable cuando se espera un contenido de datos sustancial y persistente con el tema del diseño.

5.6.1 Preocupaciones de diseño

Las preocupaciones clave incluyen la estructura persistente de los datos, el contenido de los datos, las estrategias de gestión de datos, los esquemas de acceso a los datos y la definición de metadatos.

5.6.2 Elementos de diseño

Entidades de diseño: elementos de datos, tipos y clases de datos, almacenes de datos y mecanismos de acceso.

Relaciones de diseño: asociación, usos, implementaciones. Atributos de datos, sus restricciones y relaciones estáticas entre entidades de datos, agregados de atributos y relaciones.

Atributos de diseño: propiedades de persistencia y calidad.

5.6.2.1 Atributo de datos

Descripción de los elementos de datos internos de la entidad. El atributo de datos describe el método de representación, los valores iniciales, el uso, la semántica, el formato y los valores aceptables de los datos internos. La descripción de los datos puede tener la forma de un diccionario de datos que describe el contenido, la estructura y el uso de todos los elementos de datos. La información de datos debe describir todo lo relacionado con el uso de los datos o las estructuras de datos internas por parte de esta entidad. Debe incluir especificaciones de datos como formatos, número de elementos y valores iniciales. También debe incluir las estructuras que se utilizarán para representar los datos, como estructuras de archivos, matrices, pilas, colas y particiones de memoria.

Se debe especificar el significado y el uso de los elementos de datos. Esta descripción incluye aspectos como si son estáticos o dinámicos, si se van a compartir entre transacciones, si se van a utilizar como parámetro de control o como valor, recuento de iteraciones de bucle, puntero o campo de enlace. Además, la información de los datos debe incluir una descripción de la validación de datos necesaria para el proceso.

NOTA: Este atributo de diseño se conserva por compatibilidad con la norma IEEE Std 1016-1998.

5.6.3 Ejemplos de idiomas

IDEF1X (IEEE Std 1320.2™-1998 [B19]), diagramas de clases UML (OMG [B28]).

5.7 Los patrones utilizan el punto de vista

Este punto de vista aborda las ideas de diseño (conceptos emergentes) como patrones de colaboración que involucran roles y conectores abstractos.

5.7.1 Preocupaciones de diseño

Las preocupaciones clave incluyen la reutilización a nivel de ideas de diseño (patrones de diseño), estilos arquitectónicos y plantillas de marcos.

5.7.2 Elementos de diseño

Entidades de diseño: colaboración, clase, conector, rol, plantilla de marco y patrón.

Relaciones de diseño: asociación, uso colaborativo y conector.

Atributos de diseño: nombre.

Restricciones de diseño: restricciones de colaboración.

5.7.3 Ejemplos de idiomas

Diagrama de estructura compuesta UML y una combinación del diagrama de clases UML y el diagrama de paquetes UML (OMG [B28]).

5.8 Punto de vista de la interfaz

El punto de vista de la interfaz proporciona a los diseñadores, programadores y evaluadores de información los medios para saber cómo utilizar correctamente los servicios proporcionados por un sujeto de diseño. Esta descripción incluye los detalles de las interfaces externas e internas que no se proporcionan en el SRS. Este punto de vista consta de un conjunto de especificaciones de interfaz para cada entidad.

NOTA: Las interfaces de usuario se abordan por separado.

5.8.1 Preocupaciones de diseño

La descripción de una vista de interfaz sirve como contrato vinculante entre diseñadores, programadores, clientes y evaluadores. Les proporciona un acuerdo necesario antes de proceder con el diseño detallado de las entidades. La descripción de la interfaz la utilizan los redactores técnicos para elaborar la documentación para el cliente o los clientes pueden utilizarla directamente. En este último caso, la descripción de la interfaz podría dar lugar a la producción de una vista de interfaz humana.

Los diseñadores, programadores y evaluadores suelen utilizar entidades de diseño que no han desarrollado. Estas entidades pueden reutilizarse de proyectos anteriores, contratarse de una fuente externa o producirse por otros desarrolladores. La descripción de la interfaz establece un acuerdo entre diseñadores, programadores y evaluadores sobre cómo interactuarán las entidades que cooperan. Cada descripción de la interfaz de la entidad debe contener todo lo que otro diseñador o programador necesita saber para desarrollar software que interactúe con esa entidad. Una descripción clara de las interfaces de las entidades es esencial en un desarrollo de varias personas para lograr una integración fluida y facilitar el mantenimiento.

5.8.2 Elementos de diseño

Los atributos de identificación (4.6.2.1), función (5.3.2.1) e interfaz (5.8.2.1) deben proporcionarse para todas las entidades de diseño.

5.8.2.1 Atributo de interfaz

Descripción de cómo interactúan otras entidades con esta entidad. El atributo de interfaz describe los métodos de interacción y las reglas que rigen dichas interacciones. Los métodos de interacción incluyen los mecanismos para invocar o interrumpir la entidad, para comunicarse a través de parámetros, áreas de datos comunes o mensajes y para el acceso directo a datos internos. Las reglas que rigen la interacción incluyen el protocolo de comunicaciones, el formato de datos, los valores aceptables y el significado de cada valor.

Este atributo proporciona una descripción de los rangos de entrada, el significado de las entradas y salidas, el tipo y formato de cada entrada o salida y los códigos de error de salida. En el caso de los sistemas de información, debe incluir las entradas, los formatos de pantalla y una descripción completa del lenguaje interactivo.

NOTA: Este atributo de diseño se conserva por compatibilidad con la norma IEEE Std 1016-1998

5.8.3 Ejemplos de idiomas

Lenguajes de definición de interfaz (IDL), diagrama de componentes UML (OMG [B28]). En el caso de las interfaces de usuario, la vista de interfaz debe incluir formatos de pantalla, entradas válidas y salidas resultantes. Para las entidades basadas en datos, se debe utilizar un diccionario de datos para describir las características de los datos. Aquellas entidades que son muy visibles para un usuario e involucran los detalles de cómo el cliente debe percibir el sistema deben incluir un modelo funcional, escenarios de uso, conjuntos de características detalladas y el lenguaje de interacción.

5.9 Punto de vista de la estructura

El punto de vista de la estructura se utiliza para documentar los componentes internos y la organización del tema de diseño en términos de elementos similares (recursivamente).

5.9.1 Preocupaciones de diseño

Estructura compositiva de componentes de grano grueso y reutilización de componentes de grano fino.

5.9.2 Elementos de diseño

Entidades de diseño: puerto, conector, interfaz, parte y clase.

Relaciones de diseño: conectado, parte de, incluido, proporcionado y requerido.

Atributos de diseño: nombre, tipo, propósito y definición.

Restricciones de diseño: restricciones de interfaz, restricciones de reutilización y restricciones de dependencia.

5.9.3 Ejemplos de idiomas

Diagrama de estructura compuesta UML, diagrama de clases UML y diagrama de paquetes UML (OMG [B28]).

5.10 Punto de vista de la interacción

El punto de vista de interacción define estrategias para la interacción entre entidades, respecto de por qué, dónde, cómo y en qué nivel ocurren las acciones.

5.10.1 Preocupaciones de diseño

Para los diseñadores, esto incluye evaluar la asignación de responsabilidades en las colaboraciones, especialmente al adaptar y aplicar patrones de diseño; descubrimiento o descripción de interacciones en términos de mensajes entre los objetos afectados al cumplir con las acciones requeridas; y lógica de transición de estados y concurrencia para sistemas reactivos, interactivos, distribuidos, en tiempo real y similares.

5.10.2 Elementos de diseño

Clases, métodos, estados, eventos, señales, jerarquía, concurrencia, tiempo y sincronización.

5.10.3 Ejemplos

Diagrama de estructura compuesta UML, diagrama de interacción UML (OMG [B28]).

5.11 Punto de vista de la dinámica de estados

Los sistemas reactivos y los sistemas cuyo comportamiento interno es de interés utilizan este punto de vista.

5.11.1 Preocupaciones de diseño

Dinámica de sistemas que incluye modos, estados, transiciones y reacciones a eventos.

5.11.2 Elementos de diseño

Entidades de diseño: evento, condición, estado, transición, actividad, estado compuesto, estado de submáquina, región crítica y disparador.

Relaciones de diseño: parte de, interno, efecto, entrada, salida y adjunto a.

Atributos de diseño: nombre, finalización, activo, inicial y final.

Restricciones de diseño: condiciones de protección, concurrencia, sincronización, estado invariante, restricción de transición y protocolo.

5.11.3 Ejemplos de idiomas

Diagrama de máquina de estados UML (OMG [B28]), diagrama de estados de Harel, tabla de transición de estados (matriz), autómatas, red de Petri.

5.12 Punto de vista del algoritmo

La descripción detallada del diseño de las operaciones (como métodos y funciones), los detalles internos y la lógica de cada entidad de diseño.

5.12.1 Preocupaciones de diseño

El punto de vista del algoritmo proporciona detalles que necesitan los programadores y analistas de algoritmos con respecto al rendimiento espacio-temporal y la lógica de procesamiento antes de la implementación, y para ayudar en la producción de planes de pruebas unitarias.

5.12.2 Elementos de diseño

Estos deben incluir las descripciones de los atributos para la identificación, el procesamiento (5.12.1) y los datos de todas las entidades de diseño.

5.12.3 Atributo de procesamiento

Descripción de las reglas que utiliza la entidad para lograr su función. El atributo de procesamiento describe el algoritmo que utiliza la entidad para realizar una tarea específica y sus contingencias. Esta descripción es un refinamiento del atributo de función y es el nivel de refinamiento más detallado para la entidad.

Esta descripción debe incluir los tiempos, la secuencia de eventos o procesos, los requisitos previos para la iniciación del proceso, la prioridad de los eventos, el nivel de procesamiento, los pasos reales del proceso, las condiciones de la ruta y los criterios de retorno o terminación del bucle. El manejo de contingencias debe describir la acción que se debe tomar en caso de condiciones de desbordamiento o en caso de falla de la verificación de validación.

NOTA: Este atributo de diseño se conserva por compatibilidad con la norma IEEE Std 1016-1998.

5.12.4 Ejemplos

También se pueden utilizar tablas de decisiones y diagramas de flujo; lenguajes de diseño de programas, "pseudocódigo" y lenguajes de programación (reales).

5.13 Punto de vista de los recursos

El propósito del punto de vista de recursos es modelar las características y la utilización de los recursos en un tema de diseño.

5.13.1 Preocupaciones de diseño

Las preocupaciones clave incluyen la utilización de recursos, la contención de recursos, la disponibilidad y el rendimiento.

5.13.2 Elementos de diseño

Entidades de diseño: recursos, políticas de uso.

Relaciones de diseño: asignación y usos.

Atributos de diseño: identificación (4.6.2.1), recurso (5.13.2.1), medidas de rendimiento (como rendimiento, tasa de consumo).

Restricciones de diseño: prioridades, bloqueos, restricciones de recursos.

5.13.2.1 Atributo de recursos

Descripción de los elementos que utiliza la entidad y que son externos al diseño. El atributo de recursos identifica y describe todos los recursos externos al diseño que necesita esta entidad para realizar su función. Este atributo debe especificar las reglas de interacción y los métodos para utilizar el recurso.

Este atributo proporciona información sobre elementos como dispositivos físicos (impresoras, particiones de disco, bancos de memoria), servicios de software (bibliotecas matemáticas, servicios del sistema operativo) y recursos de procesamiento (ciclos de CPU, asignación de memoria, buffers).

El atributo de recursos debe describir las características de uso, como el tiempo de procesamiento en el que se deben adquirir los recursos y el tamaño, que debe incluir la cantidad y los tamaños físicos del uso del búfer. También debe incluir la identificación de posibles condiciones de carrera y bloqueo, así como las facilidades de gestión de recursos.

NOTA: Este atributo de diseño se conserva por compatibilidad con la norma IEEE Std 1016-1998.

5.13.3 Ejemplos

Woodside [B37], Diagrama de clases UML, Lenguaje de restricciones de objetos UML (OMG [B28]).

Anexo A

(informativo)

Bibliografía

[B1] Abran, A., y JW Moore (editores), Guía del conocimiento en ingeniería de software, edición 2004. IEEE Press, 2005.

[B2] Arlow, J., e I. Neustadt, Patrones empresariales y MDA: creación de un mejor software con patrones arquetípicos y UML. Addison-Wesley, 2004.

[B3] Coplien, J., Diseño multiparadigma para C++. Addison-Wesley, 1998.

[B4] D'Souza, D., y A. Wills, Objetos, componentes y marcos con UML. Addison-Wesley 1999.

[B5] Douglass, BP, Doing Hard Time: Desarrollo de sistemas en tiempo real con UML, objetos, marcos y patrones. Addison-Wesley, 1999.

[B6] Douglass, BP, UML en tiempo real. Tercera edición, Addison-Wesley, 2004.

[B7] Evitts, P., Un lenguaje de patrones UML. Sams, 2000.

[B8] Fayad, ME y RE Johnson (editores), Marcos de aplicación específicos de dominio. Wiley, 2000.

[B9] Feldmann, CG, Guía práctica para la reingeniería de procesos de negocios utilizando IDEF0. Dorset House Publishing, 1998.

[B10] Gamma, E., et al., Patrones de diseño: elementos de software orientado a objetos reutilizable. Addison-Wesley, 1995.

[B11] Gomma, H., Diseño de líneas de productos de software con UML. Addison-Wesley, 2005.

[B12] Graham, I., Métodos orientados a objetos. Addison-Wesley, 2001.

[B13] IEEE 100™, Diccionario autorizado de términos de normas IEEE, séptima edición. Nueva York: Institute of Electrical and Electronics Engineers, Inc.⁷,

[B14] IEEE Std 610.12™-1990, Glosario estándar IEEE de terminología de ingeniería de software.

[B15] IEEE Std 830™-1998, Práctica recomendada del IEEE para especificaciones de requisitos de software.

[B16] IEEE Std 1012™-2004, Estándar IEEE para verificación y validación de software.

[B17] IEEE Std 1028™-2008, Estándar IEEE para revisiones y auditorías de software.

[B18] IEEE Std 1320.1™-1998, Estándar IEEE para lenguaje de modelado funcional: sintaxis y semántica para IDEF0.

[B19] IEEE Std 1320.2™-1998, Lenguaje de modelado conceptual estándar IEEE: sintaxis y semántica para IDEF1X97 (IDEFObject).

[B20] IEEE Std 1471™-2000, Práctica recomendada por IEEE para la descripción arquitectónica de sistemas con uso intensivo de software.⁹

[B21] IEEE Std 12207™-2008 Ingeniería de sistemas y software: procesos del ciclo de vida del software.¹⁰

⁷ Las publicaciones del IEEE están disponibles en el Instituto de Ingenieros Eléctricos y Electrónicos, 445 Hoes Lane, Piscataway, NJ 08854, EE. UU. (<http://standards.ieee.org/>).

⁸ Las normas o productos IEEE a los que se hace referencia en el Anexo A son marcas comerciales propiedad del Instituto de Electricidad y Electrónica. Ingenieros, Incorporados.

⁹ La norma IEEE Std 1471-2000 también se conoce como ISO/IEC 42010:2007.

¹⁰ La norma IEEE Std 12207-2008 también se conoce como ISO/IEC 12207:2008.

[B22] Proyecto de norma ISO, Gráficos conceptuales, 2001.11

[B23] ISO/IEC 13568:2002, Tecnología de la información—Notación de especificación formal Z—Sintaxis, sistema de tipos y semántica.12

[B24] ISO/IEC 13817-1:1996, Tecnología de la información—Lenguajes de programación, sus entornos e interfaces de software de sistema—Método de desarrollo de Viena—Lenguaje de especificación—Parte 1: Lenguaje base.

[B25] ISO/IEC 15289:2006, Ingeniería de sistemas y software—Contenido de los productos de información del proceso del ciclo de vida de sistemas y software (Documentación).

[B26] Kruchten, P., El proceso racional unificado. Addison-Wesley, 2000.

[B27] Nuseibeh, B., J. Kramer y A. Finkelstein, "Un marco para expresar las relaciones entre múltiples vistas en la especificación de requisitos", IEEE Transactions on Software Engineering, 20(10) pp. 760–773, 1994.

[B28] OMG formal/2007-11-02, Lenguaje de modelado unificado (OMG UML), Superestructura, versión 2.1.2, noviembre de 2007.

[B29] OMG formal/2007-11-04, Lenguaje de modelado unificado (OMG UML), Infraestructura, versión 2.1.2, noviembre de 2007.

[B30] Page-Jones, M., Guía práctica para el diseño de sistemas estructurados, segunda edición. Prentice Hall, 1988.

[B31] Parnas, DL y PC Clements, "Un proceso de diseño racional: cómo y por qué fingirlo", IEEE Transactions on Software Engineering, 12(7), 1986.

[B32] Ross, DT, "Análisis estructurado: un lenguaje para comunicar ideas", IEEE Transactions on Software Engineering, 1977.

[B33] Ross, DT, JB Goodenough y CA Irvine, "Ingeniería de software: proceso, principios y objetivos", COMPUTER 8(5) (mayo de 1975): 17–27

[B34] Rumbaugh, J., I. Jacobson y G. Booch, Manual de referencia UML, segunda edición, Addison-Wesley, 2005.

[B35] Shon, D., El practicante reflexivo. Basic Books, 1983.

[B36] Weiss, D. y C. Lai, Ingeniería de líneas de productos de software. Addison-Wesley, 1999.

[B37] Woodside, CM, "Un modelo de tres vistas para la ingeniería de rendimiento de software concurrente", IEEE Transactions on Software Engineering, 21(9), 1995.

[B38] Yourdon, E., y L. Constantine, Diseño estructurado. Prentice Hall, 1979.

¹¹ Disponible en <http://www.jfsowa.com/cg/cgstand.htm>.

¹² Las publicaciones ISO/IEC están disponibles en la Secretaría Central de ISO, Case Postale 56, 1 rue de Varembe, CH-1211, Genève 20, Switzerland/Suisse (<http://www.iso.ch/>). Las publicaciones ISO/IEC también están disponibles en los Estados Unidos en Global Engineering Documents, 15 Inverness Way East, Englewood, Colorado 80112, USA (<http://global.ihs.com/>). Las copias electrónicas están disponibles en los Estados Unidos en American National Standards Institute, 25 West 43rd Street, 4th Floor, New York, NY 10036, USA (<http://www.ansi.org/>).

Anexo B

(informativo)

Descripción del lenguaje de diseño conforme

Este anexo define un formato uniforme para describir lenguajes de diseño. Cualquier lenguaje de diseño puede documentarse en términos de una serie de características. Estas características se han elegido para facilitar la selección de lenguajes de diseño en la selección y definición de puntos de vista (Cláusula 4). El formato tiene un formato textual estructurado destinado a permitir su aplicación tanto por humanos como por máquinas.

Se prevé que los proveedores de lenguajes de diseño (ya sean comerciales, industriales, de investigación o experimentales) podrán documentar el uso previsto del lenguaje de diseño utilizando este formato. Esta documentación permitirá a los diseñadores revisar más fácilmente las propiedades de ese lenguaje de diseño para su uso en un SDD porque los atributos capturados aquí coinciden con las consideraciones que se deben tener en cuenta al definir un punto de vista de diseño de acuerdo con las Cláusulas 4 y 5 de esta norma.

Toda descripción de lenguaje de diseño bien formada debe especificar el nombre del lenguaje de diseño.

Ejemplos de nombres de lenguajes de diseño:

"IDEF0" "Máquina de estados UML"

Toda descripción de lenguaje de diseño bien formulada debe tener una referencia a la definición del lenguaje de diseño. Esta puede ser una referencia a una norma u otro documento de definición.

Ejemplo de referencia de lenguaje de diseño:

Estándar IEEE 1302.1, Lenguaje de modelado unificado OMG, v1.4, septiembre de 2001

Toda descripción de lenguaje de diseño bien formulada debe contener una identificación de una o más preocupaciones de diseño que se puedan expresar mediante este lenguaje de diseño. Los diseñadores pueden utilizar esta información para elegir lenguajes de diseño adecuados para implementar puntos de vista de diseño seleccionados dentro de un SDD.

Ejemplos de preocupaciones de diseño:

Funcionalidad, confiabilidad (para ejemplos adicionales consulte la cláusula 5).

Toda descripción de lenguaje de diseño bien formada debe identificar cada tipo de entidad de diseño definido por el lenguaje de diseño.

Ejemplos de elementos de diseño:

Estado, Transición, Evento (para ejemplos adicionales consulte la Cláusula 5).

Toda descripción de lenguaje de diseño bien formada debe identificar cada tipo de atributo de entidad de diseño y la entidad de diseño que lo define.

Ejemplos de atributos de diseño:

transitionLabel definida por: Transition; guardCondition definida por: Transition (para ejemplos adicionales, consulte la Cláusula 4 y la Cláusula 5).

Toda descripción de lenguaje de diseño bien formada debe identificar los tipos de relación de entidades de diseño que forman parte del lenguaje de diseño. Primero, se nombra el tipo de relación. Luego, se identifican los tipos de entidades de diseño que participan en la relación.

Ejemplos de relaciones de diseño:

Participantes en subActividades: 2 o más Actividades (para ejemplos adicionales consulte la Cláusula 5).

Anexo C

(informativo)

Plantillas para un SDD

La plantilla de la Figura C.1 muestra una forma posible de organizar y formatear un SDD conforme a los requisitos de la Cláusula 4.

Pieza frontal	
Fecha de emisión y estado	
Organismo emisor	
Paternidad literaria	
Historial de cambios	
Introducción	
Objetivo	
Alcance	
Contexto	
Resumen	
Referencias	
Glosario	
Cuerpo	
Partes interesadas identificadas y preocupaciones de diseño	
Punto de vista del diseño 1	
Vista de diseño 1	
...	
Punto de vista del diseño n	
Vista de diseño n	
Fundamento del diseño	

Figura C.1—Tabla de contenidos de un SDD