Software Design Description

IEEE STD 1016 —Systems Design— Software Design Descriptions

- Clause 1 defines the scope and purpose of the standard.
- Clause 2 provides definitions of terms used within the context of the standard.
- Clause 3 provides a framework for understanding SDDs in the context of their preparation and use.
- Clause 4 describes the required content and organization of an SDD.
- Clause 5 defines several design viewpoints for use in producing SDDs.
- Annex A provides a bibliography.
- Annex B defines how a design language to be used in an SDD is to be described in a uniform manner.
- Annex C contains a template for organizing an SDD conforming to the requirements of this standard.

2

1. Overwiev (Scope, Purpose, Intended audiance, Conformance)

- This standard specifies requirements on the information content and organization for software design descriptions (SDDs)
- This standard is intended for use in design situations in which an explicit SDD is to be prepared.
- These situations include traditional software design and construction activities leading to an implementation as well as "reverse engineering" situations where a design description is to be recovered from an existing implementation.
- Applicability is not restricted by size, complexity, or criticality of the software.
- This standard is intended for technical and managerial stakeholders who prepare and use SDDs.
- An SDD conforms to this standard if it satisfies all of the requirements in Clause 4 and Clause 5

3

2. Definitions

- **design element**: An item occurring in a design view that may be any of the following: design entity, design relationship, design attribute, or design constraint.
- **design concern**: An area of interest with respect to a software design. (functionality, reliability, performance, maintainability ..)
- **design view**: A representation comprised of one or more design elements to address a set of design concerns from a specified design viewpoint: SDD is organized using design views.
- **design viewpoint**: The specification of the elements and conventions available for constructing and using a design view.
- **design overlay**: A representation of additional, detailed, or derived design information organized with reference to an existing design view.

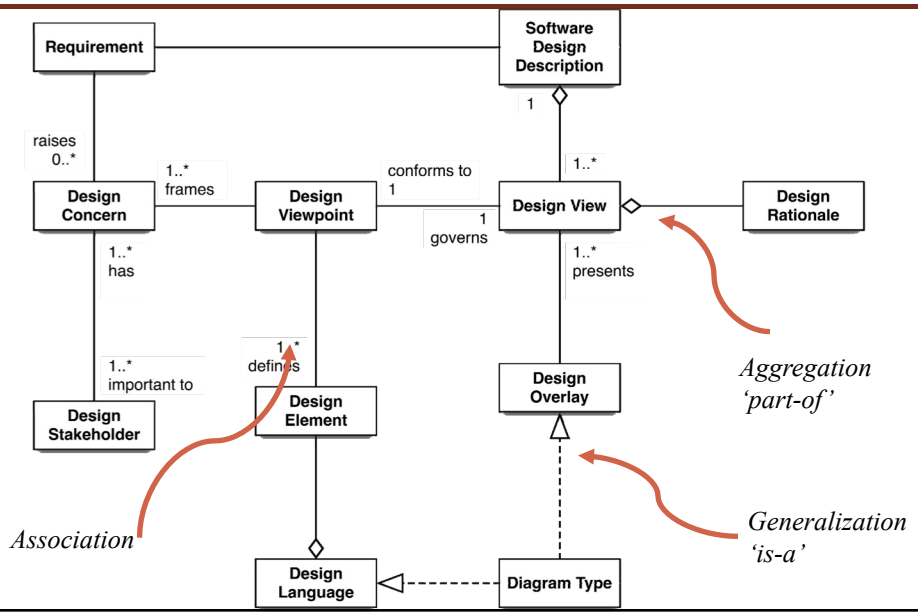
4

… definitions – Design Elements

- **design entity:** An element of a design view that is structurally, functionally, or otherwise distinct from other elements, or plays a different role relative to other design entities.
- **design relationship:** Element of a design view that names a connection or correspondence between design entities.
- **design constraint:** An element of a design view that names and specifies a rule or restriction on a design entity, design attribute, or design relationship.
- **design attribute:** An element of a design view that names a characteristic or property of a design entity, design relationship, or design constraint.

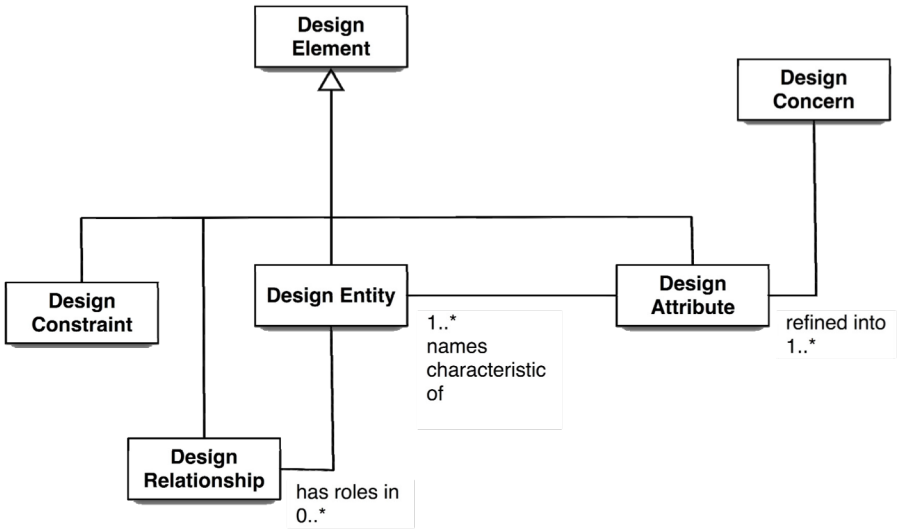
5

3. Conceptual Model



6

Conceptual Model: Design elements



7

4. Design description information content 4.1 Introduction

- The required contents of an SDD are as follows:
  - Identification of the SDD
  - Identified design stakeholders
  - Identified design concerns
  - Selected design viewpoints, each with type definitions of its allowed design elements and design languages
  - Design views
  - Design overlays
  - Design rationale

8

4.2 SDD identification

- An SDD shall include the following descriptive information:
  - Date of issue and status
  - Scope
  - Issuing organization
  - Authorship (responsibility or copyright information)
  - References
  - Context
  - One or more design languages for each design viewpoint used
  - Body
  - Summary
  - Glossary
  - Change history

9

4.3 Design stakeholders and their concerns

- An SDD shall identify the design stakeholders for the design subject.
- An SDD shall identify the design concerns of each identified design stakeholder.
- An SDD shall address each identified design concern.

10

4.4 Design views

- An SDD shall be organized into one or more design views.
- Each design view in an SDD shall conform to its governing design viewpoint.
- The purpose of a design view is to address design concerns pertaining to the design subject, to allow a design stakeholder to focus on design details from a specific perspective and effectively address relevant requirements.
- Each design view shall address the design concerns specified by its governing design viewpoint.

11

4.5 Design viewpoints

- For each design view in an SDD, there shall be exactly one design viewpoint governing it.
- Each design viewpoint shall be specified by:
  - Viewpoint name;
  - Design concerns that are the topics of the viewpoint;
  - Design elements, defined by that viewpoint, specifically the types of design entities, attributes, relationships, and constraints introduced by that viewpoint or used by that viewpoint (which may have been defined elsewhere). These elements may be realized by one or more design languages;
  - Analytical methods or other operations to be used in constructing a design view based upon the viewpoint, and criteria for interpreting and evaluating the design; and
  - Viewpoint source (e.g., authorship or citation), when applicable.

12

4.6 Design elements

- A design element is any item occurring in a design view. A design element may be any of the following subcases: design entity, design relationship, design attribute, or design constraint.
- Each design element in the SDD shall have a name (4.6.2.1), a type (4.6.2.2), and any contents.
- The type of each design element shall be introduced within exactly one design viewpoint definition.
- A design element may be used in one or more design views.

13

4.6.1 Design entities

- Design entities capture key elements of a software design.
- Each design entity shall have a name, a type, and purpose.
- Examples of design entities include, but are not limited to, the following: systems, subsystems, libraries, frameworks, abstract collaboration patterns, generic templates, components, classes, data stores, modules, program units, programs, and processes.

14

4.6.2 Design attributes

- A design attribute names a characteristic or property of a design element (which may be a design entity, design constraint, or a design relationship) and provides a statement of fact about that design element.
- All design attributes declared by a design viewpoint shall be specified.

15

Design attributes

- 4.6.2.1 Name attribute
  - The name of the element. Each design element shall have an unambiguous reference name. The names for the elements may be selected to characterize their nature. This will simplify referencing and tracking in addition to providing identification.
- 4.6.2.2 Type attribute
  - A description of the kind of element. The type attribute shall describe the nature of the element. It may simply name the kind of element, such as subsystem, component, framework, library, class, subprogram, module, program unit, function, procedure, process, object, persistent object, class, or data store.

16

Design attributes

- 4.6.2.3 Purpose attribute
  - A description of why the element exists. The purpose attribute shall provide the rationale for the creation of the element.
- 4.6.2.4 Author attribute
  - Identification of designer. The author attribute shall identify the designer of the element.

17

4.6.3 Design relationships

- A design relationship names an association or correspondence among two or more design entities. It provides a statement of fact about those design entities.
- Each design relationship in an SDD shall have a name and a type. A design relationship shall identify the design entities participating in the relationship.
- For example
  - OOAD relationships include encapsulation, generalization, specialization, composition, aggregation, realization, and instantiation.

18

4.6.4 Design constraints

- A design constraint is an element of a design view that names a rule or restriction imposed by one design element (the source) upon another design element (the target), which may be a design entity, design attribute, or design relationship.
- Each design constraint in an SDD shall have a name and a type.

19

4.7 Design overlays

- A design overlay is used for presenting additional information with respect to an already-defined design view.
- Each design overlay shall be uniquely named and marked as an overlay.
- Each design overlay shall be clearly associated with a single viewpoint.

20

4.8 Design rationale

- Design rationale captures the reasoning of the designer that led to the system as designed and the justifications of those decisions.
- Design rationale may take the form of commentary, made throughout the decision process and associated with collections of design elements.
- The only required design rationale is use of the purpose attribute.

21

4.9 Design languages

- Design languages are selected as a part of design viewpoint specification.
- A design language may be selected for a design viewpoint only if it supports all design elements defined by that viewpoint.
- Design languages shall be selected that have:
  - A well-defined syntax and semantics; and
  - The status of an available standard or equivalent defining document.
- Only standardized and well-established (i.e., previously defined and conveniently available) design languages shall be used in an SDD.

22

5. Design viewpoints

Design viewpoint	Design concerns	Example design languages
Context (5.2)	Systems services and users	IDEF0, UML use case diagram, Structured Analysis context diagram
Composition (5.3) Can be refined into new viewpoints, such as: functional (logical) decomposition, and run-time (physical) decomposition.	Composition and modular assembly of systems in terms of subsystems and (pluggable) components, buy vs. build, reuse of components	Logical: UML package diagram, UML component diagram, Architecture Description Languages, IDEF0, Structure chart, HIPO Physical: UML deployment diagram
Logical (5.4)	Static structure (classes, interfaces, and their relationships) Reuse of types and implementations (classes, data types)	UML class diagram, UML object diagram
Dependency (5.5)	Interconnection, sharing, and parameterization	UML package diagram and component diagram
Information (5.6) with data distribution overlay and physical volumetric overlay	Persistent information	IDEF1X, entity-relation diagram, UML class diagram
Patterns (5.7)	Reuse of patterns and available Framework template	UML composite structure diagram
Interface (5.8)	Service definition, service access	Interface definition languages (IDL), UML component diagram
Structure (5.9)	Internal constituents and organization of design subjects, components and classes	UML structure diagram, class diagram
Interaction (5.10)	Object communication, messaging	UML sequence diagram, UML communication diagram
State dynamics (5.11)	Dynamic state transformation	UML state machine diagram, statechart (Harel's), state transition table (matrix), automata, Petri net
Algorithm (5.12)	Procedural logic	Decision table, Warnier diagram, JSP, PDL
Resources (5.13) May be refined into resource based	Resource utilization	UML Real-time Profile, UML class diagram, UML Object Constraint

23

5.4 Logical viewpoint

- The purpose of the Logical viewpoint is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships.
- 5.4.1 Design concerns
  - The Logical viewpoint is used to address the development and reuse of adequate abstractions and their implementations. The main concern is the proper choice of abstractions and their expression in terms of existing types.

24

… 5.4 Logical viewpoint

- 5.4.2 Design elements
  - Design entities: class, interface, power type, data type, object, attribute, method, association class, template, and namespace.
  - Design relationships: association, generalization, dependency, realization, implementation, instance of, composition, and aggregation.
  - Design attributes: name, role name, visibility, cardinality, type, stereotype, redefinition, tagged value, parameter, and navigation efficiency.
  - Design constraints: value constraints, relationships exclusivity constraints, navigability, generalization sets, multiplicity, derivation, changeability, initial value, qualifier, ordering, static, pre-condition, post-condition, and generalization set constraints.
- 5.4.3 Example languages
  - UML class diagrams and UML object diagrams (showing objects as instances of their respective classes)

25

5.8 Interface viewpoint

- The Interface viewpoint provides information designers, programmers, and testers the means to know how to correctly use the services provided by a design subject. This description includes the details of external and internal interfaces not provided in the SRS. This viewpoint consists of a set of interface specifications for each entity.
- 5.8.1 Design concerns
  - An Interface view description serves as a binding contract among designers, programmers, customers, and testers.
  - Each entity interface description should contain everything another designer or programmer needs to know to develop software that interacts with that entity.

26

…Interface viewpoint

- 5.8.2 Design elements
  - The attributes for identification, function, and interface should be provided for all design entities.
- 5.8.2.1 Interface attribute
  - A description of how other entities interact with this entity. The interface attribute describes the methods of interaction and the rules governing those interactions. Methods of interaction include the mechanisms for invoking or interrupting the entity, for communicating through parameters, common data areas or messages, and for direct access to internal data. The rules governing the interaction include the communications protocol, data format, acceptable values, and the meaning of each value.

27

5.10 Interaction viewpoint

- The Interaction viewpoint defines strategies for interaction among entities, regarding why, where, how, and at what level actions occur.
- 5.10.1 Design concerns
  - For designers.
- 5.10.2 Design elements
  - Classes, methods. states, events, signals, hierarchy, concurrency, timing, and synchronization.
- 5.10.3 Examples
  - UML sequence diagram.

28

Template for an SDD

- Fronts piece
  - Date of issue and status
  - Issuing organization
  - Authorship
  - Change history
- Introduction
  - Purpose
  - Scope
  - Context
  - Summary
- References
- Glossary
- Body
  - Identified stakeholders and design concerns
  - Design viewpoint 1
  - Design view 1
  - …
  - Design viewpoint n
  - Design view n
  - Design rationale

29