Software Design Description

SDD

Software Design Description

- A representation of a software system that is used as a medium for communicating software design information
- A blueprint or model of the software system
- created to facilitate analysis,planning,implementation,and decision making

Purpose

- Shows how the software system will be structured to satisfy the requirements identified in the software requirement specification
- Translation of requirements into a description of the software structure,software components,interfaces and the data necessary for the implementation phase
- Detailed blueprint of the implementation activity

Design Description information content

The required contents of an SDD:

- **Identification of the SDD**
- **Identified design stake holders**
- **Identified design concerns**
- **Selected design viewpoints**
- **Design views**
- **Design overlays**
- **Design rationale**

SDD Identification

SDD include the following descriptive information:

- Date of issue and status
- Scope
- Issuing organisation
- Authorship(responsibility or copyright information)
- References
- Context

SDD Identification

- One or more design languages for each design viewpoint used
- Body
- Summary
- Glossary
- Change history

Table of contents for an SDD

Design stakeholders and their concerns

Design stakeholder

An individual,organization,or group having an interest in the design of some software item.

Example: users,developers,software designers etc.

SDD shall

- identify the design stakeholders for the design subject
- identify the design concerns of each identified design stakeholder
- address each identified design concern

Design view

- A representation comprised of one or more design elements
- address one or more design concerns
- **Example** of typical design concern include functionality,reliability,performance and maintainability.
- SDD is organised into design views
- Governed by design viewpoint

Design viewpoint

The specification of the elements and conventions available for constructing and using a design view.

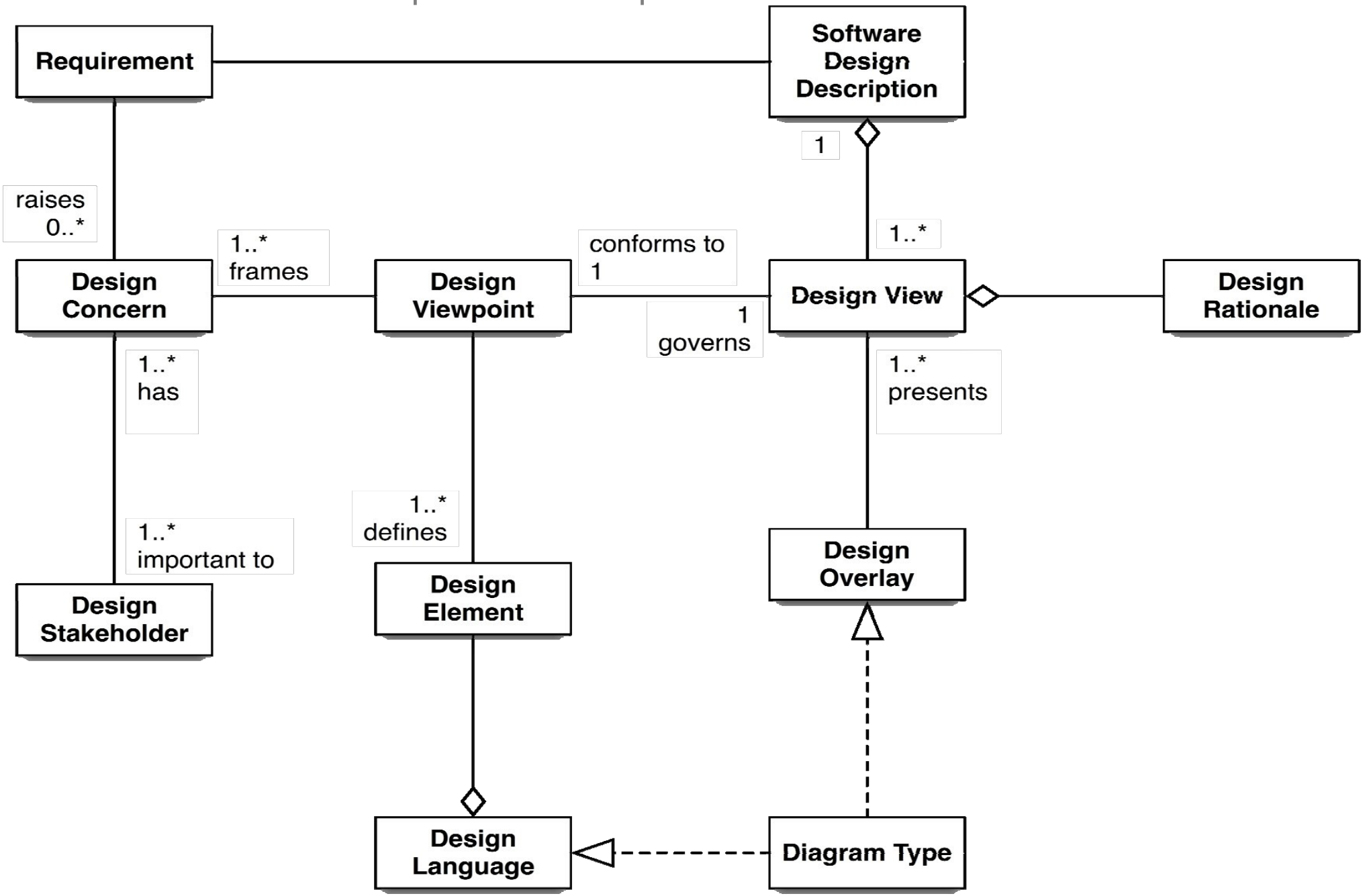
Example:

A viewpoint can introduce familiar design elements such as functions,input,and outputs;these elements are used to create a functional view.

Each design viewpoint shall be specified by

- Viewpoint name
- Design concerns that are the topics of the viewpoint
- Design elements
- Analytical methods
- Viewpoint source(eg:authorship or citation),when applicable

Conceptual model-top view



Context viewpoint

- Depicts the services provided by a design subject with reference to an explicit context
- Purpose is to identify a design subjects offered sevices,its users and other stakeholders to establish the system boundary
- **Example** of design languages:
  - IDEF0,UML case diagram,Structured Analysis Context diagram

Composition viewpoint

- Describes the way the design subject is structured into constituent parts and establishes the roles of those parts
- Design concern is the composition and modular assembly of systems in terms of subsystems and components
- **Example** of design languages are
  - UML Package diagram,UML Component diagram

Logical viewpoint

- Used to address development and reuse of adequate abstractions and their implementations.
- **Design concern**
  - Static structure(classes,interfaces,and their relationships)
  - Reuse of types and implementations
- **Example** of design languages:UML class diagram,UML object diagram

Dependency viewpoint

- Provides overall picture of the design subject inorder to assess the impact of the requirements or design changes
- **Design concern**:

  interconnection,sharing and parameterization
- **Example** of design languages:UML package diagram and component diagram

Information viewpoint

- Applicable when there is a substantial persistent data content expected with the design subject
- **design concerns**
  - persistent data structure
  - data content
  - data access schemes
  - definition of metadata
- **Example**:Entity-relation diagram,UML class diagram

Patterns use viewpoint

- This viewpoint addresses design ideas as collaboration patterns involving abstracted roles and connectors
- Design concern include reuse of patterns and available Framework template
- **Example**:UML composite structure diagram

Interface viewpoint

- Provides information desigers,programers and testers the means to know how to correctly use the services provided by the design subject
- Design concern include
  - service definition
  - service access
- **Example**:UML component diagram

Structure viewpoint

- Used to document the internal constituents and organization of the design subject in terms of elements(recursively)
- **Example** of design languages: UML structure diagram,class diagram

Interaction viewpoint

- Defines strategies for interaction among entities
- **Design concern** include
  - object communication
  - messaging
- **Example**:UML sequence diagram,UML communication diagram

State dynamics viewpoint

- Reactive systems and systems whose internal behaviour is of interest use this viewpoint
- **Design concern**:dynamic state transformation
- **Example**:UML state machine diagram,Petri net

Algorithm viewpoint

- The detailed design description of operations,the internal details and logic of each design entity
- **Design concern** is the procedural logic
- **Example**:decision table,Warnier diagram

Resource viewpoint

- Purpose is to model the characterics and utilisation of resources in a design subject
- **Design concern** :resource utilisation
- **Example**:UML Real-time profile,UML class diagram,UML Object Constraint Language

Design elements

Any item occuring in the design view that may be any of the following:

- Design entity
- Design attributes
- Design relationships
- Design constraints

Design entities

- An element of a design view that is structurally, functionally different from other elements,or plays a different role to other design entities
- **Example**:systems,subsystems,libraries,classes, programs,modules,processes
- key element of a software design
- Each design entity shall have a name,type and purpose

Design attributes

Names a characteristic or property of a design element (which may be a design entity,design constraint,or a design relationship)

- Name attribute
- Type attribute
- Purpose attribute
- Author attribute

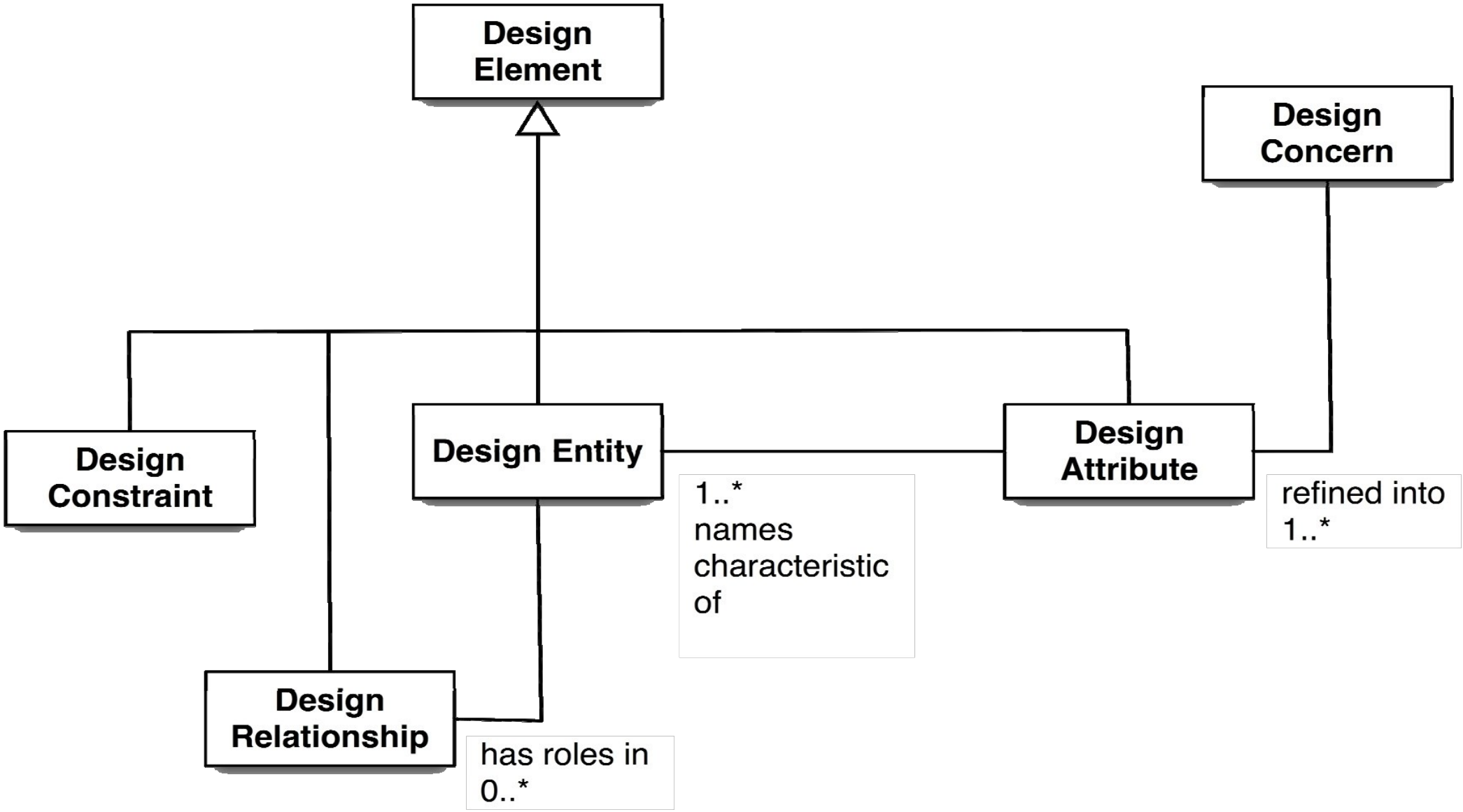
Design relationships

- Association or correspondence among two or more design entities.
- Provides a statement of fact about those design entities

Design constraints

- Element of a design view that names a rule or restriction imposed by one design element(source) upon another design element(target)
- Each design constraint shall have a name and type.

Conceptual model:design elements



Design overlays

- Used for representing additional information with respect to an already defined design view

  Each design overlay shall be

  - Uniquely named
  - Associated with a single viewpoint

Design rationale

- Captures the reasoning of the designer that led to the system as designed and the justification of those decisions

  For eg: It may take the form of

  - Commentary,made throughout the decision process
  - Design issues raised and addressed in response to design concerns
  - Design options considered
  - Trade offs evaluated etc.

Design Languages

- 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 the viewpoint
- Design languages shall be selected that have
  - Well defined syntax and semantics
  - The status of an available standard or equivalent defining document
- Eg of standard languages include Unified Modelling Language (UML), Vienna Definition Method(VDM)

References

- IEEE Standard for SDD 1016-2009
  (Revision of IEEE Std 1016-1998)