Disposition: Resolved - DRAFT

OMG Issue No: 18719

Title: View and Viewpoint Construction Limitations (from Issue 18391 c), d), e) and g))

Source:

Jet Propulsion Lab (Chris Delp – Christopher.L.Delp@jpl.nasa.gov)

INCOSE (Sanford A. Friedenthal – safriedenthal@gmail.com)

Summary:

An important capability of a model based approach is the ability to automatically generate Views of the information from the model to support specific stakeholder Viewpoints. These Views may include the presentation of the modeling information in multiple forms such as diagrams, tables, or entire documents captured in different formats (e.g., MS Word, html, ppt, video). The View and Viewpoint constructs in SysML were included to aid in the automatic generation of Views, by enabling the specification of the View information and its presentation to address the stakeholder concerns. The View generation is generally implemented by other rendering applications.

At the SE DSIG meeting on June 18, 2012 in Cambridge, several individuals presented and demonstrated common practices for View generation from a model that are providing value to end users. The presentations are available from [the Cambridge SE DSIG meeting page](http://syseng.omg.org/syseng_info.htm#Cambridge-meeting-2012). The practices required the users and vendors to further extend View and Viewpoint in different ways to overcome inherent limitations in order to leverage their respective View generation capabilities. The lack of a standard approach limits interchange and requires that each user and vendor include their unique extensions.

The specific limitations of View and Viewpoint are described below. For background, the Viewpoint and View descriptions in the SysML specification v1.3 currently read as follows:

**Viewpoint:** A Viewpoint is a specification of the conventions and rules for constructing and using a view for the purpose of addressing a set of stakeholder concerns. The languages and methods for specifying a view may reference languages and methods in another viewpoint. They specify the elements expected to be represented in the view, and may be formally or informally defined. For example, the security viewpoint may require the security requirements, security functional and physical architecture, and security test cases.

**View:** A View is a representation of a whole system or subsystem from the perspective of a single viewpoint. Views are allowed to import other elements including other packages and other views that conform to the viewpoint.

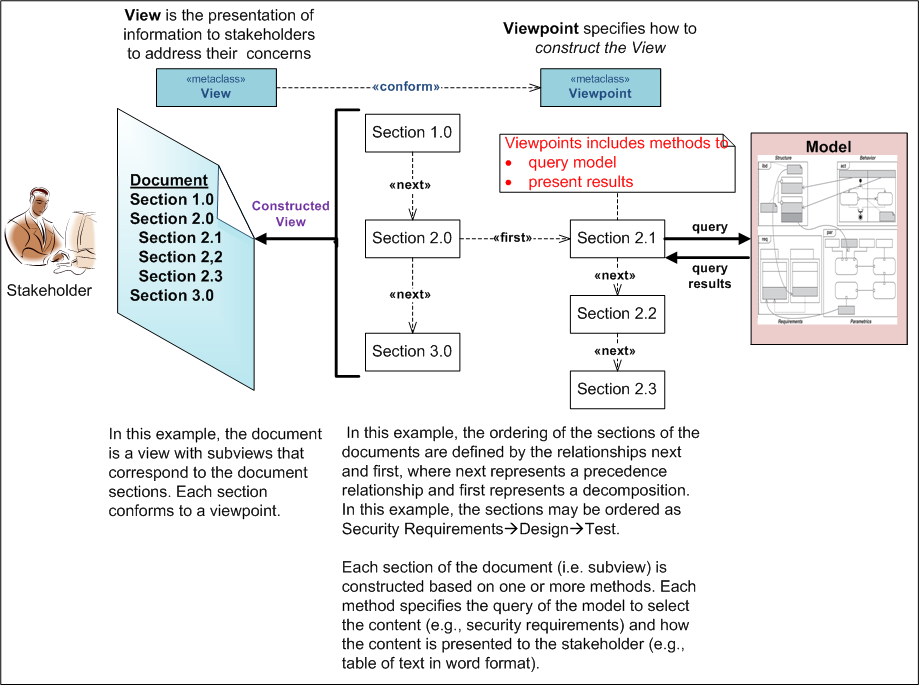
Based on the above descriptions, the Viewpoint specifies how to construct a View, and the View is a representation that conforms to this specification.

Some of the limitations that have been identified include the following:

1. **Viewpoint description limitations.** The current viewpoint description should be clarified to note that it should specify the presentation of the information as well as the information itself. This may require additional viewpoint properties to enable the specification of the form and format of the information. The form of the data in this context refers to how the information is presented such as data values that are in tabular form or a plot. The format of the data in this context refers to the file format that is used for the rendering application.
2. **View import limitations.** The current View description says “Views are allowed to import other elements including other packages and other Views that conform to the Viewpoint”. View also includes a constraint that ‘A view can only own element import, package import, comment and constraint elements’. This concept of importing model elements into a package is not a sufficient means for constructing Views. The relationship between the view and the model elements should reflect the concept that the View can be constructed by defining operations to query models and other sources of data, and perform other operations on the information to present it in a form that is useful to the stakeholders.
3. **Other view construction limitations.** A View conforming to a Viewpoint may be constructed from different sets of information that may be rendered as an entire document, a part of a document, a set of powerpoint slides or an individual slide, a video or series of videos, or other form. A typical example may be a security View that represents security requirements, design, and verification information. This requires the View to be constructed from sub-views, and that these sub-views must be ordered in a particular way to present to the user. An example would be the ordering of sections in a document, where each section represents a subview which in-turn represents selected information.

A current limitation is the inability to express the ordering and general organization of the View and corresponding subviews that comprise the View (Note: this is a structural ordering and not a temporal ordering). Some of the current approaches have addressed this limitation by including a dependency relationship between the subviews. The relationships can express a precedence relationship (i.e.., next) and a decomposition relationship (i.e., first). A simple example of how these relationships are used to construct a View that is presented to the stakeholder as a document is included below.

In the above example, different subviews correspond to the different sections of the document that comprise the View. For example, some text with a table of information from one part of the model may appear in Section 1, and some other text with a diagram that represents other model information may appear in Section 2. Each section of the document may require different viewpoints to specify the query and presentation. There is currently no way to describe that a View that conforms to a Viewpoint contains multiple subviews with the relationships as indicated in the figure. There is a need to create a View that contains subviews that are related to one another with the types of relationships indicated (e.g., first, next). (Note: It is anticipated that the View and subviews should be reusable, and may require additional metadata).

****

In the example above, each section of the document corresponds to a particular subview. However, we do not want to restrict a subview so that the information cannot be distributed across multiple sections of a document, or across multiple documents.

1. **Reuse of view and viewpoint.** There needs to be sufficient expression to construct reusable definitions of view and viewpoint. These mechanisms may include composition, specialization, model libraries, and others.
2. **Other View and Viewpoint Mechanisms.** There may be additional ways to create views more directly in the model. For example, a view may correspond to a filtered subset of a set of parts on an ibd corresponding that are based on some criteria (e.g., all electrical parts). This is similar to issue 13928 called the partition construct (later referred to as element group).

Resolution:

A summary of the proposed changes in this resolution include:

1. Extend View from Class instead of from Package to enable means for referencing views from viewReference.
2. Add a presentation property to Viewpoint whose value(s) is/are the URI to a presentation format for the view
3. Establish a relationship between an instance of a view and an instance of the model that the view is intended to represent.

Revised Text:

{Precise editing instructions for applying resolution, including exact text, models, diagrams, references to be included or deleted. **NOTE: IDL should be shown in Courier font**}

Disposition: Resolved

1. **Modify subsection 7.1 by removing the following text from the last paragraph:**

SysML has extended the concept of view and viewpoint from UML to be consistent with the *IEEE 1471 standard*. In particular, a viewpoint is a specification of rules for constructing a view to address a set of stakeholder concerns, and the view is intended to represent the system from this viewpoint. This enables stakeholders to specify aspects of the system model that are important to them from their viewpoint, and then represent those aspects of the system in a specific view. Typical examples may include an operational, manufacturing, or security view/viewpoint.

1. **Add the following subsection to the overview in Chapter 7**

**7.1.1 View and Viewpoint**

Systems Engineers use SysML to make models of systems—the resulting product, the system model, is what we mean most of the time when we speak of “the model.” Along with that model, Systems Engineers may also use SysML to make a model that captures a way to describe a system—the resulting product, the viewpoint and view model, helps Systems Engineers assure that stakeholders get the understanding they need from the system model. The notion of viewpoint and view articulated in ISO-42010 is the basis for the way we build viewpoint and view models in SysML. SysML has extended the concept of view and viewpoint from UML to be consistent with the *ISO-42010* *standard*. Typical examples may include an operational, manufacturing, or security view/viewpoint.

The viewpoint and view model can also be thought of as a description model – distinct from the notion of a system model. One viewpoint and view model exposes elements of one or more system models. In particular, a viewpoint is a specification of rules for constructing a view to address a set of stakeholder concerns, and the view is intended to represent the system from this viewpoint. This enables stakeholders to specify aspects of the system model that are important to them from their viewpoint, and then represent those aspects of the system in a specific view.

The viewpoint describes the point of view of a set of stakeholders by framing the concerns of the stakeholders along with the method for producing a view that addresses those concerns. The method describes the expectation of what stakeholder(s) wish to see exposed from the model, how the stakeholder wishes the information to be structured and presented, and in what kind of artifact the stakeholder wants to consume the information. In other words, the method is the set of rules that describe how the view should express the information from the model to address the stakeholder concerns. The method can be specified as a process for producing a view and may include rules for analyzing or verifying the view content.

The view is the modeling element that represents the artifact that the stakeholder views. The view must be related to the model that contains the information and the method that produces the view. The view is used as an input to some rendering application to generate the artifact such as a document for consumption and evaluation by the stakeholder. The viewpoint method specifies how this is to be accomplished.

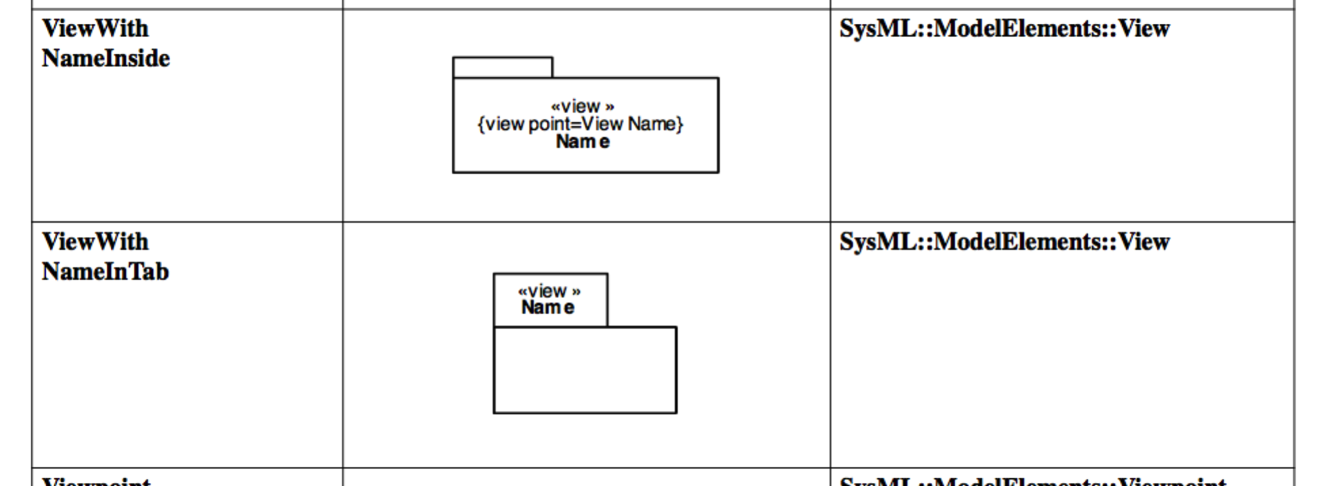
The relationship between viewpoint and view model and artifact is similar to that between system model and a particular run of output data from a simulation. While the output data is fed back into the system model, for example into equations in parametric diagrams, the artifact is “fed back” to a person or organization: the stakeholder whose concerns are expressed in the viewpoint that led to the view from which the artifact was generated.

The result is a feedback loop: stakeholder expresses concerns; concerns are collected in a viewpoint; view that conforms to viewpoint is created; model(s) of the system is related to view; artifact is generated from view using viewpoint method; stakeholder evaluates artifact against rules in method; stakeholder revises and expresses concerns and/or method until the stakeholder’s concerns are addressed.

To the extent that the stakeholder is not a model developer or architect, there is actually another feedback loop, or two-way communication exchange, involved in creating the viewpoint and view in the first place: stakeholder expresses concerns; architect captures those concerns in a viewpoint and a view; architect asks the stakeholder if the viewpoint and view are what the stakeholder intended; stakeholder revises and expresses concerns and/or method until the stakeholder is satisfied with the viewpoint and view. Alternatively, the viewpoint can be created first and one or more views created later, always based on communication between stakeholder and architect/systems engineer.

1. **In Section 7.2 Diagram Elements**,
2. **Modify Table 7.1 by replacing the 2 rows that show the view diagram elements on page 23 as follows:**

**from**

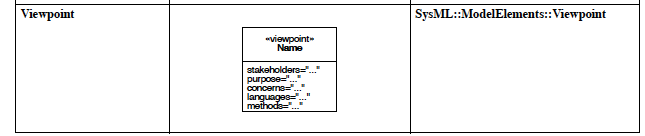
****

**to**

|  |  |  |
| --- | --- | --- |
| **View** | **(add visio for view as metaclass class)** | **SysML::ModelElements::View** |

1. **Modify Table 7.1 by replacing the viewpoint diagram element in the last row on pg 23 as follows:**

**From:**

****

**To:**

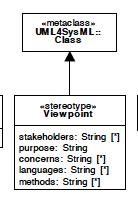
|  |  |  |
| --- | --- | --- |
| **Viewpoint** | **(add visio for viewpoint)** | **SysML::ModelElements::Viewpoint** |

1. **Modify Table 7.2 by adding the expose dependency on pg 26 as follows:**

|  |  |  |
| --- | --- | --- |
| **expose** | (add visio mapping dependency metaclass) | **SysML::ModelElements::expose** |

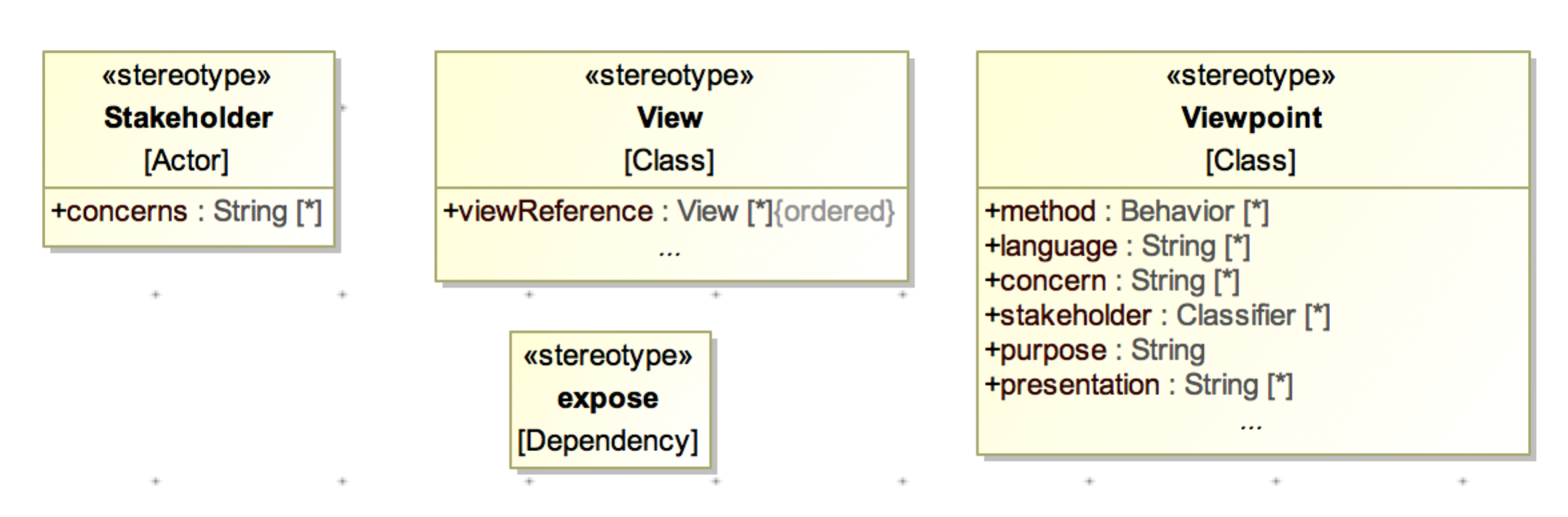
1. **In Section 7.3.2 Stereotypes,**
2. **modify Figure 7.1 by replacing the viewpoint stereotype as follows:**

**From:**

****

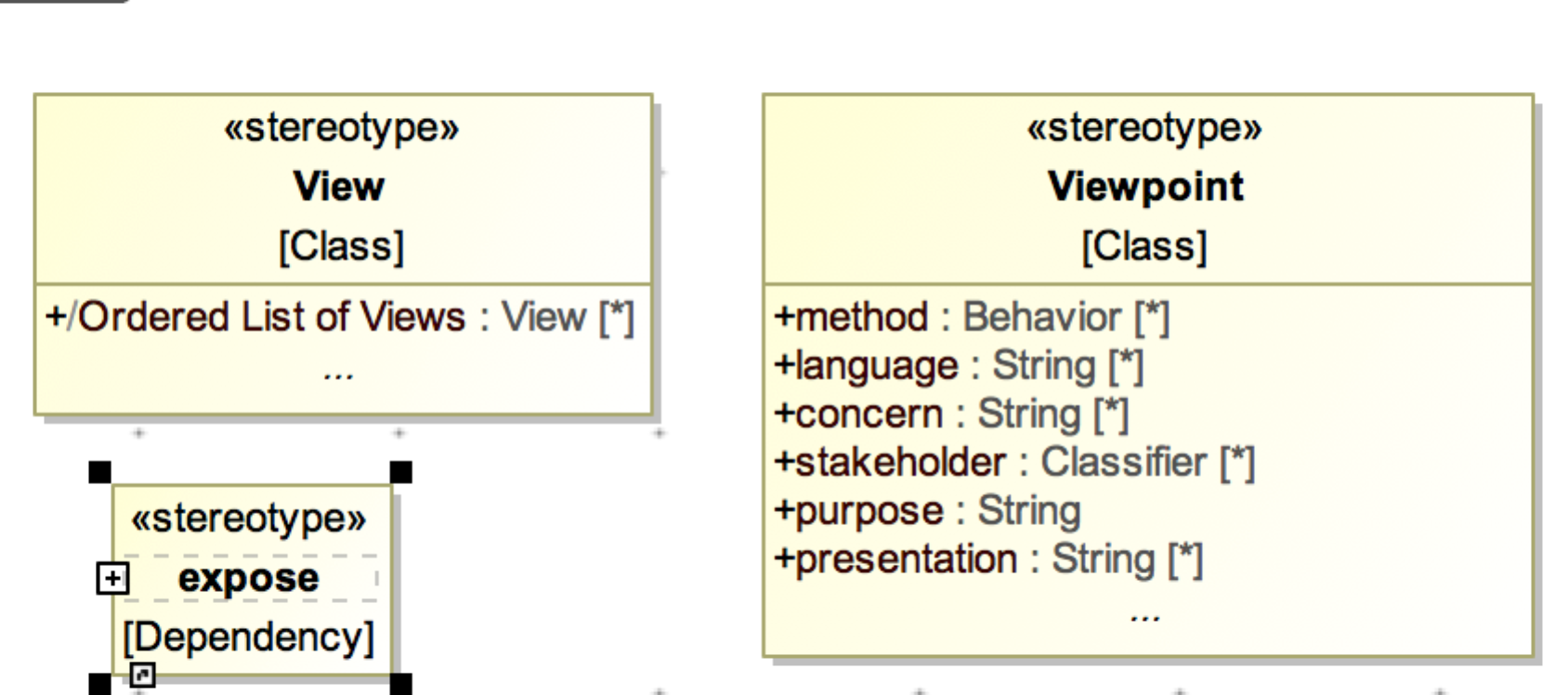
**To:**

**(add visio showing new viewpoint)**

****

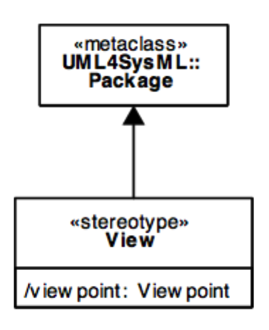
1. **modify Figure 7.1 by adding the expose stereotype as follows:**

**(add visio showing expose specializing metaclass for dependency)**

****

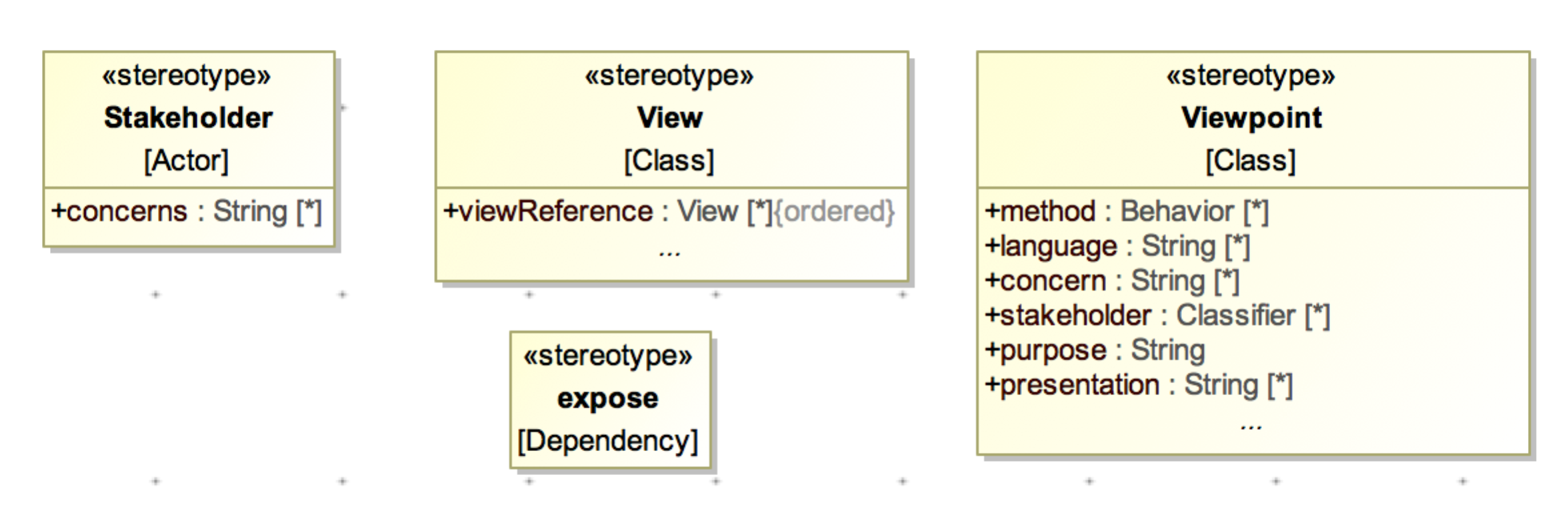
1. **modify Figure 7.1 by changing the view stereotype as follows:**

**from:**

****

**to:**

**(add visio showing view as class)**

****

1. **modify 7.3.2.4 View as follows:**

**from**

**7.3.2.4 View**

**Description**

A View is a representation of a whole system or subsystem from the perspective of a single viewpoint. Views are allowed to import other elements including other packages and other views that conform to the viewpoint.

**Attributes**

• /viewpoint: Viewpoint

The viewpoint for this View, derived from the supplier of the «conform» dependency whose client is this View.

**Constraints**

[1] A view can only own element import, package import, comment, and constraint elements.

[2] The view is constructed in accordance with the methods and languages that are specified as part of the viewpoint. SysML does not define the specific methods. The precise semantics of this constraint is a semantic variation point.

**To**

**7.3.2.4 View**

**Description**

A view is used in a viewpoint-view model to represent the artifact produced by the method. As a practical matter, artifacts are the form in which stakeholders consume the information that the method brings into being. Artifacts may include one or more documents, view graphs, animated captures of simulations or other concrete representations of information. The SysML specification does not address or limit what artifacts can be.

A view responds to the concerns in a viewpoint—we say that it conforms to that viewpoint. A view may reference other views in order to respond to concerns.

**Attributes**

• /viewpoint: Viewpoint

The viewpoint for this View, derived from the supplier of the «conform» dependency whose client is this View.

* viewReference: View [\*] {ordered}

The ordered list of «view» elements referenced by a view via shared association.

**Constraints**

[1] The view is constructed in accordance with the methods and languages that are specified as part of the viewpoint. SysML does not define the specific methods.

[2] The subordinate view attribute is restricted to listing Property elements or association roles typed by «view».

[3] A ««view» may only conform to a single ««viewpoint».

1. **Add the following attribute to the viewpoint attributes under 7.3.2.5 Viewpoint**

• presentation: String [\*]

The specifications prescribed for formatting and presenting the view as an artifact. The presentation specifications are referred to by the URI and/or name.

1. **Add the following constraint to the viewpoint constraints under 7.3.2.5 Viewpoint**

[5] The presentation property should be expressed as a URI for the presentation specifications.

1. **Add a new subsection 7.3.2.6 that provides the stereotype definition for expose as follows:**

7.3.2.6 expose

Description

The expose relationship relates a view to 1 or more model elements that represent the scope of model elements to be exposed by the view.

**Constraints**

[1] The client/source must be an element that is stereotyped by«view».

1. **Increment the remaining subsections to reflect the addition of the new subsection 7.3.2.6**

Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Name** | **Change Notes** | **Date** |
| Rev 1 | Chris Delp | Baselined changes | 5/15/13 |
| Rev 2 | Abraham Raher | Updated text in 7.1.1 | 5/15/13 |
|  | Chris Delp | Updated text in 7.1.1, changed subordinate view to viewReference | 5/15/13 |
|  |  |  |  |