The Epoch Pattern

Date: 2016-04-15
### Contents

1 Introduction ................................................................................................................. 2  
  1.1 Pattern Aims ........................................................................................................... 2  
2 Concepts ....................................................................................................................... 3  
3 Viewpoints .................................................................................................................. 4  
  3.1 Overview ................................................................................................................. 4  
  3.2 Rules ....................................................................................................................... 5  
  3.3 Epoch Definition Viewpoint (EDVp) ...................................................................... 6  
    3.3.1 Description ......................................................................................................... 7  
    3.3.2 Example ............................................................................................................. 8  
  3.4 Applicable Viewset Viewpoint (AVVp) ................................................................. 9  
    3.4.1 Description ......................................................................................................... 10  
    3.4.2 Example ............................................................................................................. 10  
  3.5 Metric Definition Viewpoint (MDVp) ..................................................................... 11  
    3.5.1 Description ......................................................................................................... 12  
    3.5.2 Example ............................................................................................................. 13  
  3.6 Metric Usage Viewpoint (MUVp) ......................................................................... 15  
    3.6.1 Description ......................................................................................................... 16  
    3.6.2 Example ............................................................................................................. 17  
4 Summary ....................................................................................................................... 17  
5 Related Patterns ........................................................................................................ 17  
6 References & Further Reading .................................................................................... 18
1 Introduction
Measurement is a key aspect of any modelling endeavour. In order to manage and control the model it is essential that we can reason about it and, in order to reason about the model we need to be able to measure in a qualitative way, quantitative way or, more practically, both.

The Epoch Pattern allows measurements and metrics to be defined and applied to the model at specific points during the evolution of the model’s life. These specific points in time are defined as Epochs and the reasoning is enabled by Measures and Metrics

1.1 Pattern Aims
The main aims of this pattern are shown in the Architectural Framework Context View (AFCV) in Figure 1.

The diagram in Figure 1 shows the context for the Epoch Pattern. The main aim of the Epoch Pattern is to understand the evolution of a system (‘Understand evolution’) by considering the evolution of the complexity of the system over time (‘Support evolution of system through time’).

There is a single high-level constraint that makes an assumption that a model of the System must exist (‘Ensure model exists’). All of the Measure and Metrics that are defined using the Epoch Pattern Views are applied to the System model, therefore if it does not exist then no Measure nor Metrics may be applied.

There are three main use cases that must be satisfied for the Epoch Pattern to be effective which are:

- ‘Define epoch’, where a point in time is identified and an Epoch is defined at that point in time.
• ‘Identify key viewpoints’, where the System Model is looked at and a number of Viewpoints are identified that have been deemed relevant for the Epoch (‘Identify relevant viewpoints’). Alongside the Viewpoints, it is also important that the Context of these Viewpoints is also known (‘Identify contexts’).

• ‘Measure model’, which includes defining Measure and Metrics (‘Define general Metrics’) and also applying them to the System Model (‘Apply Metrics’). The results of the application of the Metrics must also be presented to the modeller (‘Present results’) and then the model must be allowed to be updated or augmented based on these results (‘Augment model’). There is a constraint that is applied to the ‘Measure model’ use case which is that automation of the application of the Metrics must be possible (‘Ensure automation is possible’).

This context drives the Epoch Pattern.

2 Concepts
The main concepts covered by the Epoch Pattern are shown in the Ontology Definition View (ODV) in Figure 2.

Figure 2 - Ontology Definition View showing Epoch concepts

The Ontology for the Epoch pattern is shown in Figure 54 and describes the concepts and terminology, as follows:
The Epoch Pattern

- ‘Epoch’ that defines a point during the Life Cycle of the Project or System that will form a reference point for Project activities. An Epoch will apply to a specific baseline of the System Model, but not all baselines will be an Epoch. Epochs may be evolved from other Epochs, and may evolve into other Epochs.
- ‘Applicable Viewset’ that is a subset of the ‘System Model’ and comprises one or more ‘View’. These Views that make up the Epoch will be visualised by one or more ‘Diagram’.
- ‘System Model’ is the abstraction of a specific System that evolves during the Life Cycle of the Project or System.
- ‘View’ is a defined set of information (based on a Viewpoint) with a standard structure that may be realised with one or more ‘Diagram’. The Views make up the ‘System Model’.
- ‘Diagram’ that is a visualisation of a ‘View’ and that uses a particular notation, in the case of this book, the language will be SysML.
- ‘Node’ is any graphical node (typically a shape) that forms part of the modelling language that is used in a Diagram.
- ‘Path’ is any graphical path (typically a line) that forms part of the modelling language that is used in a Diagram.
- ‘Measure’ that is an operation performed on a ‘Diagram’ that yields a result, such as a counting of blocks on a block definition diagram, counting interactions on a sequence diagram, etc. Measures are often not shown explicitly as they rely on simple counting, rather than any complex mathematical formula.
- ‘Metric’ that uses one or more ‘Measure’ in order to provide meaningful knowledge about a System Model, for example a Metric may calculate coupling, cohesion, etc.
- ‘Equation’ that forms the definition of both the ‘Metric’ and the ‘Measure’ and comprises an ‘Equation Definition’ (for example a mathematical formula) and its set of one or more ‘Parameter’.

The Ontology Elements defined on Ontology Definition View form the basis for all of the Viewpoints that are defined as part of the Epoch Pattern.

3 Viewpoints

This section describes the Viewpoints that make up the Epoch Pattern. It begins with an overview of the Viewpoints, defines Rules that apply to the Pattern and then defines each Viewpoint.

3.1 Overview

The Epoch Pattern defines a number of Viewpoints as shown in the Viewpoint Relationship View (VRV) in Figure 3.
The Epoch Pattern

The Epoch Pattern defines four Viewpoints that are described as follows:

- The ‘Epoch Definition Viewpoint’ that identifies and describes the Epoch that is being defined.
- The ‘Applicable Viewset Viewpoint’ that identifies the subset of the System Model that is relevant to the Epoch being defined.
- The ‘Metric Definition Viewpoint’ where all of the Metrics and Measures are defined.
- The ‘Metric Usage Viewpoint’ that shows how the Metrics and Measures (defined in the Metric Definition Views) are applied to the Applicable Viewset (defined in the Applicable Viewset Views).

Each of these Viewpoints is described in more detail in the following sections. For each Viewpoint an example is also given.

3.2 Rules
Six rules apply to the four Epoch Viewpoints, as shown in the Rules Definition View (RDV) in Figure 4.
Note that the five Rules shown in Figure 56 are the minimum that are needed, and are described at a high level as follows:

- ‘Rule EP1’ – ‘The System Model must exist’. This Rule enforces the fundamental assumption that a System Model must exist as all of the Measure and Metrics that are used as part of the Epoch Pattern are applied to the System Model. Therefore, if there is no System Model, then the Measure and Metrics cannot be applied.
- ‘Rule EP2’ – ‘The System Model must have a pre-defined set of Viewpoints that form the basis for the Applicable Viewset’. This Rule follows on from EP1 as, not only must a model exist, but it must be structured into a set of Viewpoints that describe its Views. It is these Viewpoints and Views that are used to identify the Applicable Viewset.
- ‘Rule EP3’ – ‘All Measures and Metrics that are defined must relate to the Diagrams that are used to visualise Views in the Applicable Viewset’. This Rule follows on from EP1 and EP2 and states that the Measures and Metrics that are defined must be applied to Diagrams that exist in the Applicable Viewset.
- ‘Rule EP4’ – ‘The value for each parameter on each Measure must be taken directly from a Diagram, Node or Path’. This Rule follows on from EP3 and enforces the fact that each parameter for a Measure must be taken directly from a Diagram, Node or Path. When combined with Rule EP3, it ensures that the Measures are valid according to the Diagrams, Nodes or Paths, and that these themselves are taken directly from the Applicable Viewset.
- ‘Rule EP5’ – ‘The value for each parameter on each Metric must be taken directly from a Measure, Diagram, Node or Path’. This Rule is similar to EP4 except here it is the Metrics that are being enforced. In the case of Metrics, their parameters must be taken directly from a Diagram, Node or Path, but may also be taken from a Measure.

Note that the five rules shown in Figure 4 are the minimum that are needed. Others could be added if required.

3.3 Epoch Definition Viewpoint (EDVp)

The aims of the Epoch Definition Viewpoint are shown in the Viewpoint Context View in Figure 5.
The diagram in Figure 5 shows that the main aim of the Epoch Definition Viewpoint is to ‘Support evolution of system over time’ that includes ‘Define epoch’. This is broken down further by the following three Use Cases:

- ‘Identify epoch’, where a specific point during the Life Cycle of the Project or System is identified.
- ‘Define epoch characteristics’, where the properties associated with the Epoch are defined, such as ‘Date’, ‘Version’ and ‘Model ref’.
- ‘Define relationships to other epochs’, which entails looking at other epochs in the Life Cycle and identifying relationship between them and the new epoch. This will allow the evolution of the model to be explored.

It should be stressed that the aim of this Viewpoint is to define the Epochs that show the evolution of the System Model over time, but not to define exactly what aspect of the System Model is being measured (such as size, complexity, etc.). The elements that are relevant for the Epoch Definition Viewpoint are shown in the next section.

3.3.1 Description

The Viewpoint Definition View (VDV) in Figure 6 shows the Ontology Elements that appear on an Epoch Definition Viewpoint.
The Epoch Pattern

The diagram in Figure 6 shows the elements from the Ontology that are relevant to the Epoch Definition Viewpoint. The Epoch Definition Viewpoint shows the Epoch being defined with its property values filled in and the relationships between them.

An Epoch may be related to other Epochs in two ways:

- By evolving from another Epoch, where there is an existing Epoch.
- By evolving into another Epoch, where there will be a future Epoch

Each Epoch is described by three properties, which are:

- ‘Baseline’ the version number for the Baseline that the Epoch represents.
- ‘Date’ the date at which the Epoch was created
- ‘Model Reference’ that refers to a specific artefact in the System model.

The number of properties defined for each Epoch is not limited to the three listed here, but these should be taken as a minimum set.

3.3.2 Example

An example View that conforms to the Epoch Definition Viewpoint is shown in Figure 7.

```
Epoch1: Epoch
  Date = July 2013
  Baseline = Version 2.2
  Model ref = Coffin escape

Epoch2: Epoch
  Date = April 2014
  Baseline = Version 3.4
  Model ref = Coffin escape
```

Figure 7 - An example of an Epoch Definition View showing two Epochs
The Epoch Pattern

The Epoch Definition View in Figure 7, realised as a block definition diagram, shows two instance specifications of Epoch, ‘Epoch 1’ and ‘Epoch 2’ that are connected together by the link.

In the example shown here, two Epochs have been identified: ‘Epoch 1’ and ‘Epoch 2’ Any number of Epochs may be shown using this Viewpoint, ranging from a single Epoch to a string of Epochs that are sequenced together. It should also be noted that the Epochs need not be connected in a linear fashion, it is also possible to have branches to show divergence in the evolution of the System Model.

3.4 Applicable Viewset Viewpoint (AVVP)

The aims of the Applicable Viewset Viewpoint are shown in the Viewpoint Context View in Figure 8.

The diagram in Figure 8 shows that the main aim of the Applicable Viewset Viewpoint is to ‘Support the evolution of system over time’ which includes ‘Identify key viewpoints’, which has three types associated with it:

- ‘Identify relevant viewpoints’, that requires the subset of the System Model that is relevant for the Epoch being defined to be identified.
- ‘Identify context’, where the Context refers to the Context of the Viewpoints that have been identified (hence the constraint between the two use cases). This is important as it is possible that the Context for those Viewpoints has evolved as time has gone on and, hence, this evolution must be captured.
- ‘Identify key views’, where these Views will be the specific instances of the Viewpoints that have been identified.

There is also a constraint, ‘Limit to existing views’ that really enforces Rules ‘Rule EP1’ and ‘Rule EP2’ from Figure 4.
3.4.1 Description
The Viewpoint Definition View (VDV) in Figure 9 shows the Ontology Elements that appear on an Applicable Viewset Viewpoint.

Figure 9 - Viewpoint Definition View showing the elements that appear on the Applicable Viewset Viewpoint (AVVp)

The Applicable Viewset Viewpoint shows how the ‘Applicable Viewset’ is related to a specific ‘Epoch’ and comprises a one or more ‘View’ and one or more ‘Viewpoint’.

In the case that the Epoch refers to the whole model, then this diagram becomes very simple as it need only show a single reference to the model, however, in many cases the Epoch will refer to a subset of the model. In this situation, it is important to which Viewpoint and specific associated Views are relevant.

3.4.2 Example
An example View that conforms to the Applicable Viewset Viewpoint is shown in Figure 10.
The Applicable Viewset View in Figure 10, realised as a SysML block definition diagram, shows the Views that are relevant to a specific Epoch. In this example, the Viewpoints are the ones that are defined by the Interface Pattern. Note how the Interface Pattern is being re-used here to define which Viewpoints are relevant to this Applicable Viewset View. As well as knowing which Viewpoints form part of the Applicable Viewset, it is essential that the actual Views are also identified. This is because it is possible and likely that neither the entire System Model, nor an entire set of instances of a specific Viewpoint will be relevant.

These Views are shown on the diagram as notes that are associated with their relevant Viewpoint and that provide a reference for each View that is relevant for each Viewpoint.

### 3.5 Metric Definition Viewpoint (MDVp)

The aims of the Metric Definition Viewpoint are shown in the Viewpoint Context View in Figure 11.
Figure 11 - Viewpoint Context View showing Metric Definition Viewpoint aims

The diagram in Figure 11 shows that the main aim of the Metric Definition Viewpoint is to ‘Support evolution of system over time’ which includes ‘Measure model’, which includes ‘Define general Metrics’ that has four types:

- ‘...for coupling’ that is a generalisation relating to complexity. This requires that general Metrics are defined that measure and calculate some aspect of the complexity between Nodes.
- ‘...for cohesion’ that is a generalisation relating to complexity. This requires that general Metrics are defined that measure and calculate some aspect of the complexity inside a Node.
- ‘... for structural diagrams’ that is a generalisation related to an aspect of the System Model. This requires that the general Metrics can be applied to the structural aspect of the System Model.
- ‘... for behavioural diagrams’ that is a generalisation related to an aspect of the System Model. This requires that the general Metrics can be applied to the behavioural aspect of the System Model.

It should be noted that the four types of measurement that are included in this diagram provide examples of the type of measurements that are possible and the diagram is not intended to show an exhaustive list. This is represented by the ‘(incomplete)’ constraint.

There is a general constraint that requires that any Metrics developed must be able to be automated (‘Ensure automation is possible’).

### 3.5.1 Description

The Viewpoint Definition View (VDV) in Figure 58 shows the Ontology Elements that appear on a Metric Definition Viewpoint.
The Metric Definition Viewpoint shows the Measures and the Metrics associated with them. Each Measure and Metric is defined here in terms of its Equations and associated Equation Definitions and Parameters.

Both the Measures and Metrics are described in the same way, buy using Equation. Each Equation is made up of a number of Equation Definitions that use a number of Parameters.

### 3.5.2 Example

An example View that conforms to the Metric Definition Viewpoint is shown in Figure 13.

---

**Figure 12 - Viewpoint Definition View showing the elements that appear on the Metric Definition Viewpoint (MDVp)**
The Metric Definition View in Figure 59, realised as a SysML block definition diagram. This View is used to show Measures and Metrics, both of which are visualised using SysML constraint blocks. The constraint block has three compartments that are used as follows:

- **Constraint block name** – this is the name of the Measure or Metric.
  - ‘constraint properties’ compartment, where the Equation Definition that represents the Measure or Metric is defined.
  - ‘parameters’ compartment, where the Parameters that are used by the Equation Definition are defined according to their type and multiplicity.

The diagram in Figure 59 shows four Metrics that are relevant to the Epoch used in this example. These Metrics are defined as follows.

- **‘Coupling’**, that calculates the coupling between different nodes on a single diagram, and that uses the ‘Average IO Metric’ [CRU80]. The equation for this Metric is:
  \[
  \text{Coupling} = \frac{\sum (Z_i \ldots Z_n)}{n}, \text{ where:}
  \]
  - \(Z_i\) is the average number of inputs and outputs for the node \(i\)
  - \(i\) is the count of the nodes on the diagram
  - \(n\) is the total number of nodes on the diagram

- **‘Average IO Metric’**, that calculates the average number of inputs and outputs shared over \(m\) nodes [CRU80]. The equation for the Metric is:
The Epoch Pattern

\[ Z = \frac{\text{SUMOF (Mj...Mn)}}{m} \text{, where:} \]

- \( M \) is the sum of the inputs and outputs shared by nodes \( j \) and \( i \)
- \( m \) is the total number of nodes to which \( i \) is joined

- ‘Structural Cohesion Metric’, that calculates the average number of operations per block in a diagram [LOR94]. The equation for the Metric is:

\[ SC = \frac{\text{SUMOF (Oi...On)}}{n} \text{, where:} \]

- \( O \) is the number of operations in a single block
- \( n \) is the number of blocks

- ‘Behavioural Cohesion Metric’, that calculates the complexity of a behavioural diagram by calculating the possible number of paths through its decision structure [MCC76]. The equation for the Metric is:

\[ V = e - n + p \text{, where:} \]

- \( e \) is the number of nodes
- \( n \) is the number of paths
- \( p \) is the number of start and end points

The Metrics that are defined here are used purely as an example of how established techniques may be applied to the Applicable Viewset for a specific Epoch. These Metrics were traditionally applied at the code level in software engineering in the original references, but have been tailored to be able to be applied at the model level – see [TUG01] for more information.

In the example shown here there are no explicit Measures being defined, as these Measures re simple counts on the diagram. For example, the ‘Behavioural Cohesion Metric’ defined here is using some simple counts that look at a diagram and identify how many nodes exist. These simple counts are the Measures.

3.6 Metric Usage Viewpoint (MUVp)

The aims of the Metric Usage Viewpoint are shown in the Viewpoint Context View in Figure 14.
The diagram in Figure 57 shows that the main aim of the Metric Usage Viewpoint is to ‘Support evolution of system over time’ which includes ‘Measure model’, which includes being able to apply the Metrics to the System Model (‘Apply Metrics’).

There is a general constraint that requires that any Metrics developed must be able to be automated (‘Ensure automation is possible’).

### 3.6.1 Description
The Viewpoint Definition View (VDV) in Figure 15 shows the Ontology Elements that appear on a Metric Usage Viewpoint.

The Metric Usage Viewpoint in Figure 15 shows how the Measures and Metrics are applied to the System Model via the Diagrams.

The Diagrams that the Measures and, hence, Metrics are applied to must have already been identified as part of the Applicable Viewset View.
3.6.2 Example

Example Views that conform to the Metric Usage Viewpoint are shown in Error! Reference source not found..

<table>
<thead>
<tr>
<th>Measure/Metric</th>
<th>Applied to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coupling (using Average IO Metric)</td>
<td>IIV [Package] Interface Identification View [Coffin Escape]</td>
</tr>
<tr>
<td></td>
<td>ICV [Package] Interface Identification View [Coffin Escape]</td>
</tr>
<tr>
<td></td>
<td>IDV [Package] Interface Definition View [Pump to Hole]</td>
</tr>
<tr>
<td></td>
<td>IDV [Package] Interface Definition View [Controller to Pump]</td>
</tr>
<tr>
<td></td>
<td>IDV [Package] Interface Definition View [Concrete to Coffin]</td>
</tr>
<tr>
<td>Structural Cohesion Metric</td>
<td>IIV [Package] Interface Identification View [Coffin Escape]</td>
</tr>
<tr>
<td></td>
<td>ICV [Package] Interface Identification View [Coffin Escape]</td>
</tr>
<tr>
<td></td>
<td>IDV [Package] Interface Definition View [Pump to Hole]</td>
</tr>
<tr>
<td></td>
<td>IDV [Package] Interface Definition View [Controller to Pump]</td>
</tr>
<tr>
<td></td>
<td>IDV [Package] Interface Definition View [Concrete to Coffin]</td>
</tr>
<tr>
<td>Behaviour Cohesion Metric</td>
<td>IBV [Package] Interface Behaviour View [Pump to Hole, normal operation]</td>
</tr>
<tr>
<td></td>
<td>IBV [Package] Interface Behaviour View [Pump to Hole, no concrete operation]</td>
</tr>
<tr>
<td></td>
<td>IBV [Package] Interface Behaviour View [Pump to Hole, switch fault operation]</td>
</tr>
<tr>
<td></td>
<td>PDV [Package] Protocol Definition View [Controller to Pump Protocol]</td>
</tr>
</tbody>
</table>

Table 1 An example of the Metric Usage View

The Metric Usage View in Error! Reference source not found. is realised as simple table. The table relates the Metrics and Measures that were defined in the Metrics Definition View to the relevant Views in the Applicable Viewset View.

4 Summary

This section has introduced and defined the set of Viewpoints that that make up the Epoch Pattern.

The Epoch Pattern allows modellers to define a set of Measures and Metrics that may then be applied to a Model, or sub-set of a Model.

It should be noted that the Epoch Pattern provides a mechanism that allows modellers to define Measures and Metrics, but the pattern itself does not define any Metrics or Measures. The pattern itself is purely definitional in nature. An example set of Metrics is shown here based on previous work on applying Metrics and Measures to Models,

5 Related Patterns

When using the Epoch Pattern, the following patterns may also be of use:

- Description Pattern.
The Epoch Pattern

- Traceability Pattern.
- Life Cycle Pattern.

6 References & Further Reading
Cruickshank, R.D. and Gaffney, J.E., Jr., "Software Design Coupling and Strength Metrics". Greenbelt, Maryland: NASA, Goddard Space Flight Centre, 1980