

# **ECP Requirements Workflow Walkthrough**

## **version 1.1**

### *Introduction*

The aim of this document is to walk you through the requirements workflow for the ECP example from the point of view of the OO analyst/designer. This example should be read in conjunction with the book “UML and the Unified Process” [Arlow].

### *References*

[Arlow] - UML and the Unified Process, Jim Arlow and Ila Neustadt, Addison Wesley, 2001, ISBN0201770601

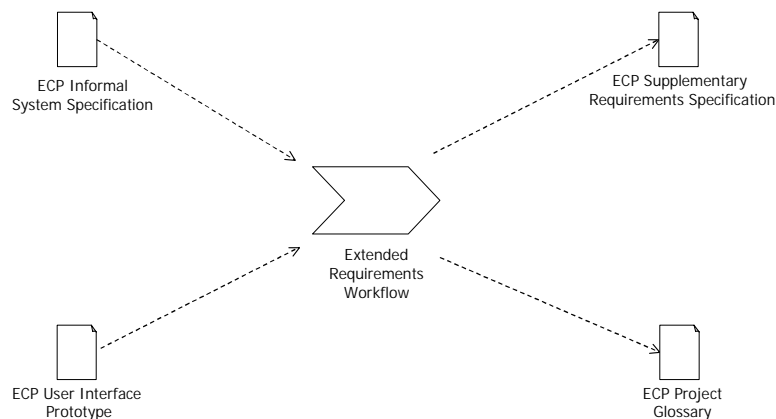
[ECPI] - ECP Informal System Specification, Clear View Training, 2002

[ECPUI] - ECP User Interface Prototype, Clear View Training, 2002

### *Inputs and outputs*

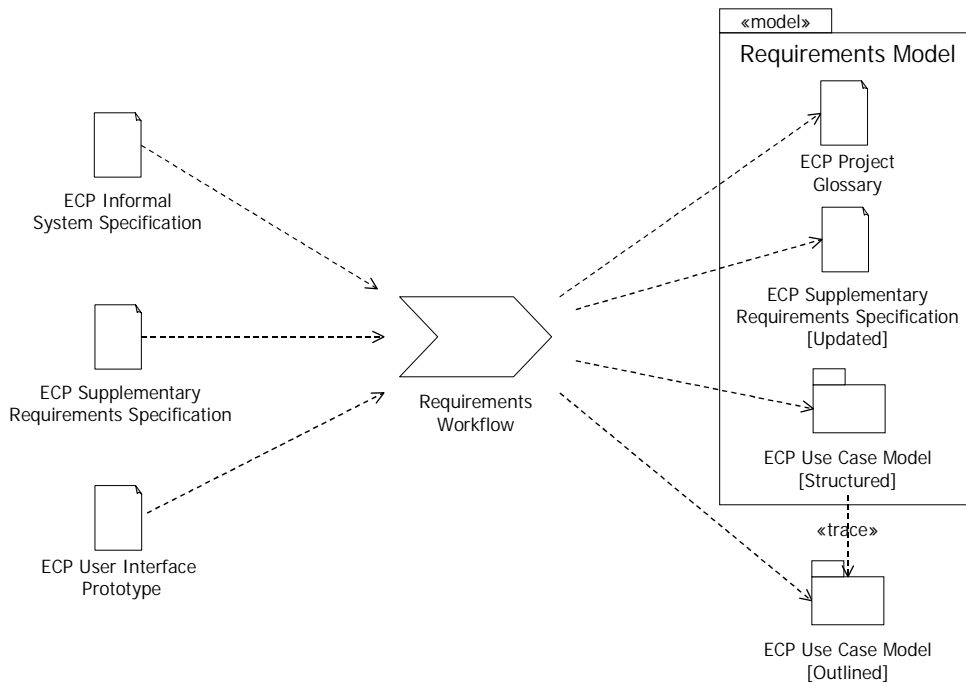
Here are the inputs and outputs to the Extended Requirements Workflow [Arlow]:

Figure 1



Here are the inputs and outputs to the UP Requirements Workflow for the ECP:

Figure 2



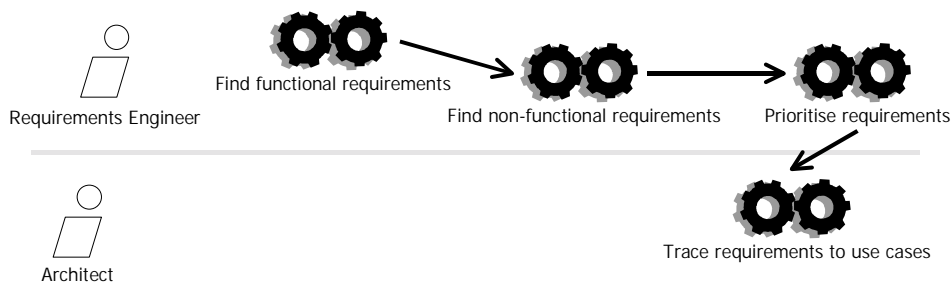
### Scope

We will be performing the extended requirements workflow [Arlow] and the standard UP requirements workflow for the Clear View Training ECP. The requirements engineering activity is based on the information contained in [ECPI] and [ECPU].

### The Extended Requirements Workflow

The Extended Requirements Workflow according to [Arlow] is shown below:

Figure 3



This workflow is made up of a number of activities:

Table 1

Activity	Purpose
Find functional requirements	Discover what behaviour the system should offer.

**Table 1**

Find non-functional requirements	Discover any constraints on the system.
Prioritize requirements	Assign a priority to each requirement.
Trace requirements to use cases	Map each requirement to all use cases that realize it.

The activity “Trace requirements to use cases” can only be performed once we have identified the use cases.

Requirements are always prioritized from according to the needs of the stakeholders. There are many possible ways of prioritizing requirements, but in this example we will use the simple MoSCoW scheme:

**Table 2**

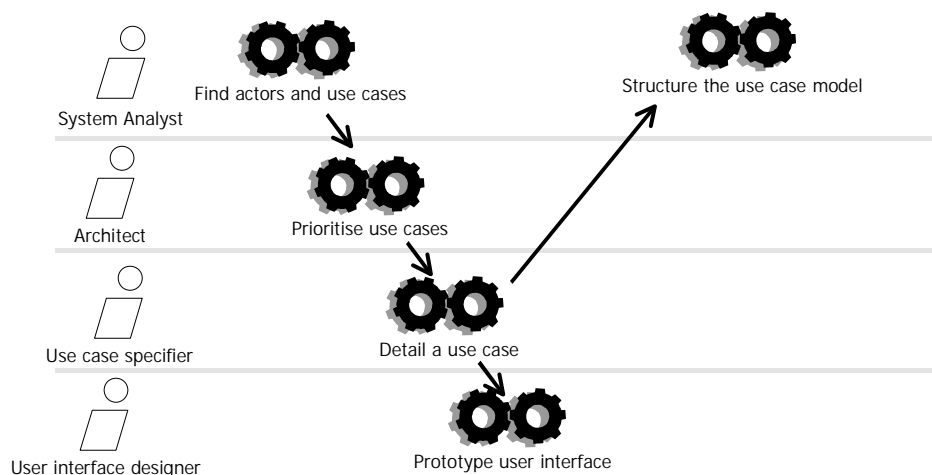
Priority	Semantics
<b>Must have</b>	Requirements that are fundamental to the system (mandatory).
<b>Should have</b>	Important requirements that may be omitted.
<b>Could have</b>	Requirements that are truly optional.
<b>Want to have</b>	Requirements that can wait for later releases of the system.

The output of this activity is the ECP Supplementary Requirements Specification (SRS).

As we create the ECP SRS we should also begin to construct the ECP Project Glossary. The project glossary is a “living document” - it should be added to as the project progresses.

### *The Requirements Workflow*

The UP Requirements Workflow is all about creating the Use Case Model.

**Figure 3**

The OO analyst/designer may be called upon to play all of the roles in this workflow except the role “User interface designer”, and so we don’t look at the activity “Prototype user interface”. User interface designer is actually a very specialised role and should really be performed by a Human Computer Interaction (HCI) expert. Many web projects suffer from letting analyst/designers and programmers handle usability issues!

Here is a summary of each of the activities in the Requirements workflow:

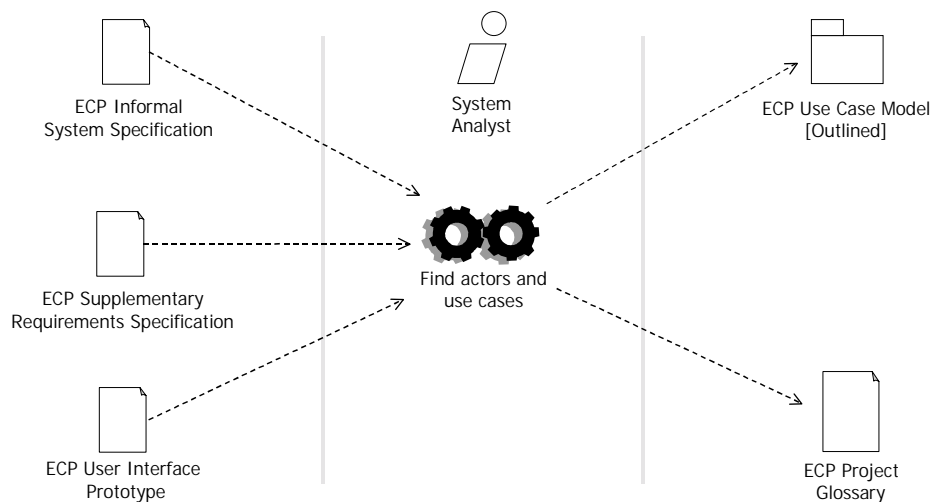
**Table 4**

Activity	Purpose
Find actors and use cases	Find where the system boundary lies by identifying who or what interacts directly with the system (actors) and what functionality the system offers to them (use cases).
Prioritize use cases	Assign priorities to the use cases to determine in which iterations they will be developed.
Detail a use case	Describe the flow of events of the use case in detail.
Prototype user interface	Create a prototype of the user interface.
Structure the use case model	Apply generalization and «include» and «extend» relationships to the use case model.

#### *Activity: Find actors and use cases*

We explain how to perform this activity in [Arlow].

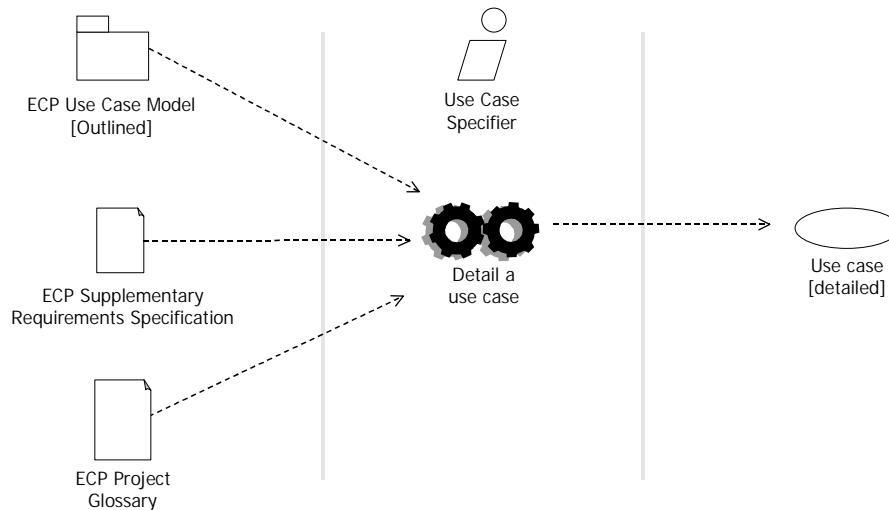
**Figure 5**



#### *Activity: Detail a use case*

This activity is about taking one of the outlined use cases and detailing its flow of events. Again, we describe this process in detail in [Arlow].

Figure 6



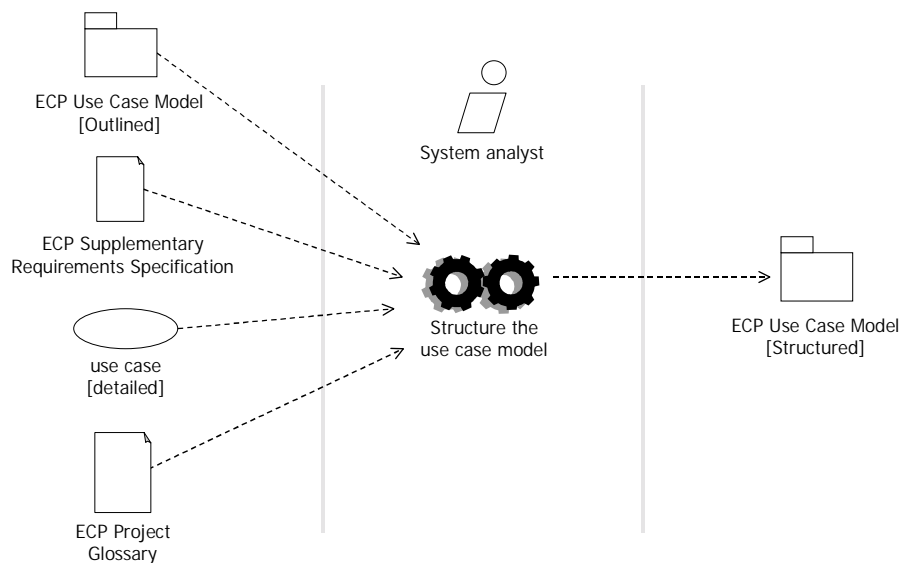
### *Activity: Structure the use case model*

As use cases are being detailed, we should structure the use case model as a parallel activity.

Structuring the use case model involves:

- Applying generalization to actors and use cases where appropriate
- Factoring common behavior out into separate use cases using «include» relationships
- Extending existing use case behavior using «extend» relationships.

Figure 7



### *Summary*

This concludes our brief walkthrough of the requirements workflow. The complete requirements model may be found on our website at [www.clearviewtraining.com/example1/index.htm](http://www.clearviewtraining.com/example1/index.htm).