UML – Business Context

Jan Kusiak, Training Services Manager
IRM Training Pty Ltd ACN 007 219 589
Suite 209, 620 St Kilda Rd, Melbourne, Vic. 3004, Australia
Tel: +613 9533 2300

Overview

"Where does UML fit?" is a common question among new (and not so new!) business analysts. We all know that the M stands for modelling but beyond this, perceptions start to differ. In its current form (V2.0) UML consists of 13 diagram types all of which provide a different view of a system. In the following extract from our **RUML**[1] course manual we'll take a brief look at which of the 13 diagrams are of most relevance for us and how they fit together.

[1] RUML – Modelling Requirements with Use Case & the UML

Introduction

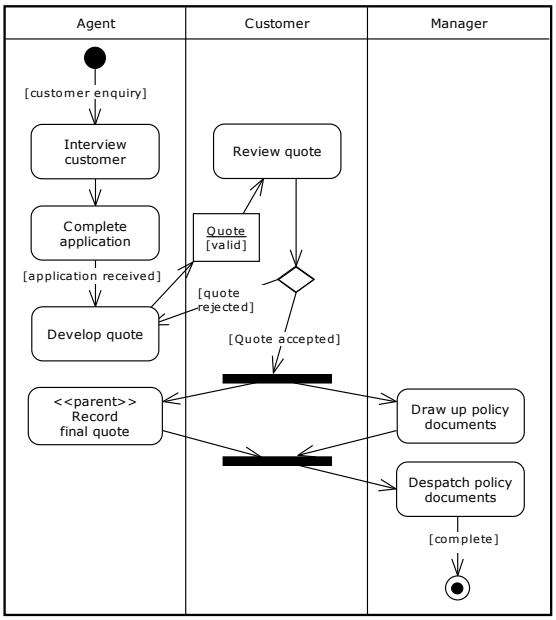
For those of you new to diagramming techniques, think of an architect's plans for a new house – there will be front, side and top views, electrical wiring and plumbing diagrams, plus specific diagrams for such things as foundations, load-bearing walls… etc.

As a business analyst we are primarily concerned with what our system (house) will do. For example we may specify that the house will have a home theatre, intercom, zoned climate control, keyless entry, etc. but we will not be doing the wiring diagram ourselves. This is done by the people who will be building the system (i.e. installing the electrics). However we can draw a diagram (a model) to show where the plasma screen, intercom and control panel should be and here is where the relevance of UML lies – we are using models (diagrams) to describe our system.

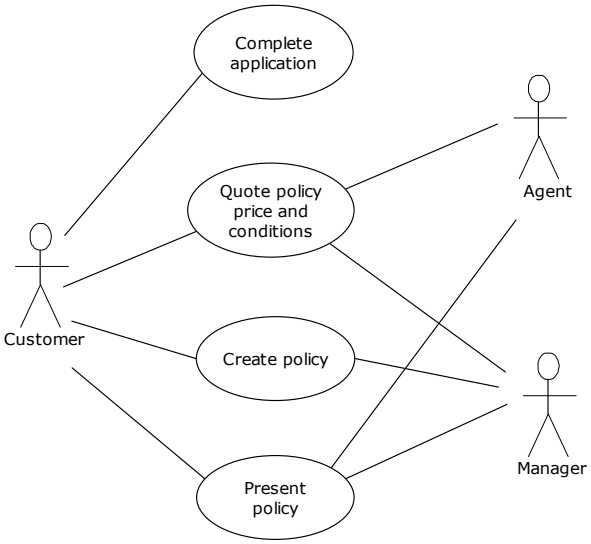
© 2007 IRM Training Pty Ltd

Which diagrams to use

Activity diagrams are used to describe the overall process. They show flows through a sequence of states with activities and sub-activities being performed. Activity diagrams can be used in much the same way as dataflow diagrams to provide a high-level view of the system or process. At a lower level, activity diagrams can also be used to model the detail flow inside a use case.

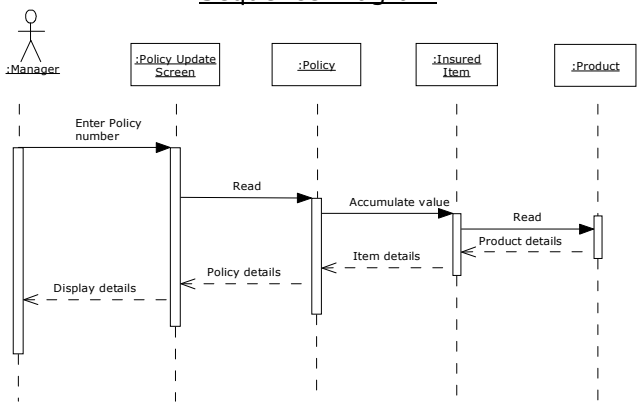


Use Cases are the main medium of communication with business users about their requirements – the business functions. They describe everyday system behaviour (events) such as a credit card purchase or an insurance policy application. They describe behaviour both for a given event (pay for goods by credit card) for alternatives (card needs authorisation) and for exceptions (credit card declined). Use cases can be both textual and diagrammatic.

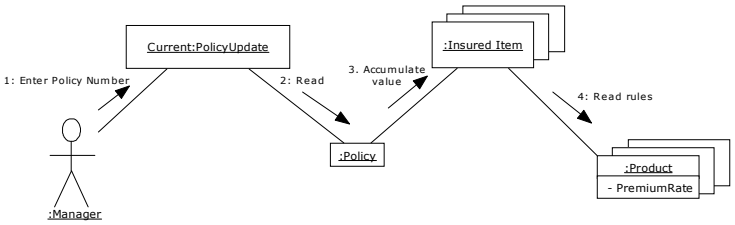


Interaction (Sequence & Communication) Diagrams are used to document the communications that must go on between the user and the system, and internally between system components. Sequence diagrams show behaviour based on time and flow of messages. Communication diagrams show the flow of messages between objects and classes.

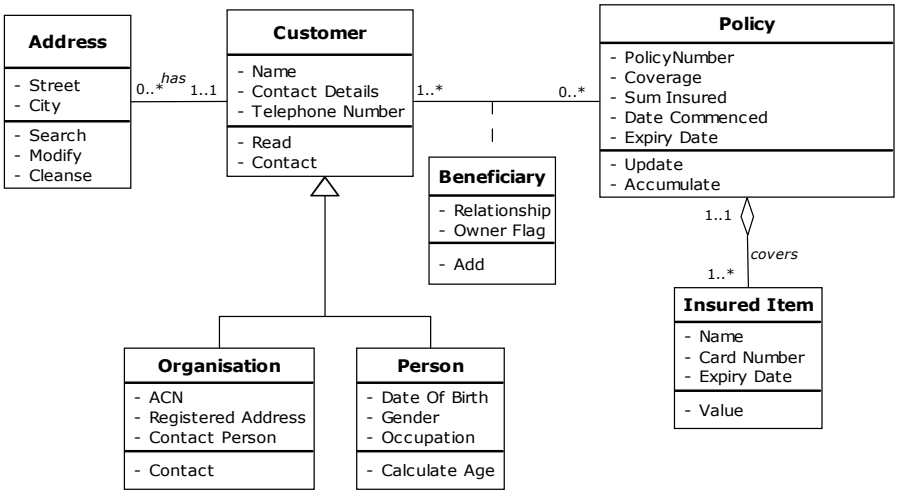
Sequence Diagram



Communication Diagram



Class Diagrams - used to group together things that have the same attributes and the same behaviour. Class diagrams can be used to model data by only showing the attribute layer and the relationships. However this is not true data modelling as the natural structure of the data is not shown. You will often find entity relationship diagrams used in conjunction with UML to give a true data modelling representation.

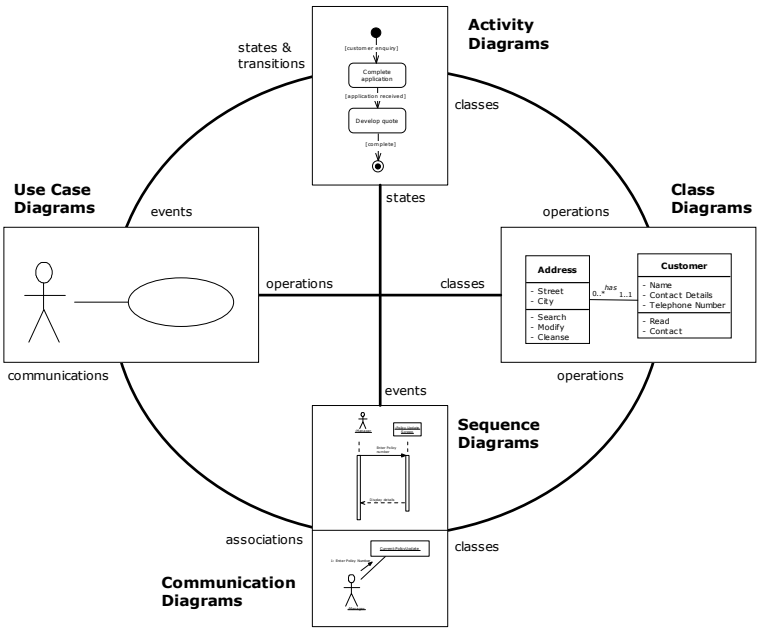


Business Perspective – Interaction between diagrams

Before UML came along business analyst used structured techniques to describe business functionality. The diagrams and techniques of choice were:

- dataflow diagrams to model the business process
- mini specs to define the process logic (business rules)
- entity relationship diagrams to model the underlying data
- a data dictionary to define the data in the data model

Together these gave us a comprehensive representation of what was required – the resulting document was called a *Functional Specification*. These techniques are still widely used but now UML provides another option and we can use the following conceptual diagram to give us a business perspective of our system.



UML techniques are not incompatible with structured techniques and both show the requirements in different ways. Structured techniques differ in that they separate the data and process, considering each one independently, before bringing them together to complete the model. Object-oriented techniques consider the data and process as closely inter-dependent. They are even held together in the same notation within each class or object.

© 2007 IRM Training Pty Ltd. All rights reserved.

Send feedback and comments to: training@irm.com.au

You may use this article in your newsletter or internal document free of charge, provided that you do not alter it in any way and include all copyright notices.