Multiple and Hierarchical Refinement in OPM

Natalia Kwasnikowska   Jan Van den Bussche
Hasselt University and Transnational University of Limburg, Belgium

August 14, 2009

The OPM v1.01[1] model already has a basic notion of refinements in the form of pairs of accounts (α^+, α^-). The meaning of such a pair is that account view α^+ (here called refiner account) provides more detail about the process than account view α^- (here called refinee account). We will also refer to this as the detailed view and the high-level view. What is lacking is a methodology for structuring accounts so that multiple refinements, about different parts of the overall process, can be independently switched on and off in a view. Such a methodology should also accommodate hierarchical refinement (only hinted at in OPM v1.01), in which a refinement can have itself certain parts that can be further refined. Combining multiple and hierarchical refinement yields a powerful modeling methodology. The aim of this note is to propose such a methodology.

The gist of our proposal is that we evolve from the simple set of refinement pairs from OPM v1.01 to a more complex structure that consists of a hierarchy on refinee accounts, together with a one-to-one correspondence between refinee accounts and refiner accounts. The result will be that every refinee account will have many possible refinements, given by certain unions of refiners and refinees situated lower in the hierarchy. Exactly which such unions are allowed will be made clear in the following.

1 Multiple refinement

Consider the OPM graph of Figure 1. The idea is that the vertical chain on the left describes the overall process. The bottom subgraph $A_1 \to P_3 \to A_2$ on the right gives more detail about artifact A; the top subgraph $Q_1 \to B_3 \to Q_2$ on the right gives more detail about process Q. In order to model these refinements we need to be able to address many different subgraphs:

- The part of the graph that is common to all different levels of detail is the subgraph induced by the basic processes P_1 and P_2 and the basic artifacts

[1] OPM v1.01 at http://eprints.ecs.soton.ac.uk/16148/1/opm-v1.01.pdf

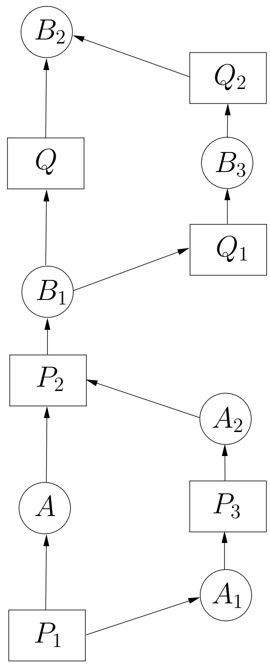

Figure 1: Multiple refinement.

B_1 and B_2. We cover this part by an account α_{root} which can be considered the "root account".

- The bottom refinee subgraph $P_1 \to A \to P_2$, which can be refined to the refiner subgraph $P_1 \to A_1 \to P_3 \to A_2 \to P_2$. We cover the refinee by an account α_A^- and we cover the refiner by an account α_A^+.
- Similarly, the top refinee subgraph $B_1 \to Q \to B_2$, which can be refined to the refiner subgraph $B_1 \to Q_1 \to B_3 \to Q_2 \to B_2$. We cover the refinee by account α_Q^- and the refiner by α_Q^+.

All the above accounts are referred to as "refinement accounts" because they are used to model refinements. The general idea of the root account is that it contains the parts of the graph that are not involved in any refinement.

Note that we have a one-to-one correspondence $\alpha^- \mapsto \alpha^+$ between refinee accounts and refiner accounts. This is a general feature of our methodology.

Note also that account views α_{root}, α_A^- and α_A^+ overlap nicely in the two nodes P_1 and P_2, and similarly, that α_{root}, α_Q^- and α_Q^+ overlap precisely in B_1 and B_2. Making sure we have proper overlaps is, in our methodology, an essential task.

Moreover, we need to explicitly declare the hierarchical relationship between the root account and her two refinee accounts. This can be done by declaring a set H of hierarchy pairs, in this case consisting of the two pairs $(\alpha_{\text{root}}, \alpha_A^-)$ and $(\alpha_{\text{root}}, \alpha_Q^-)$.

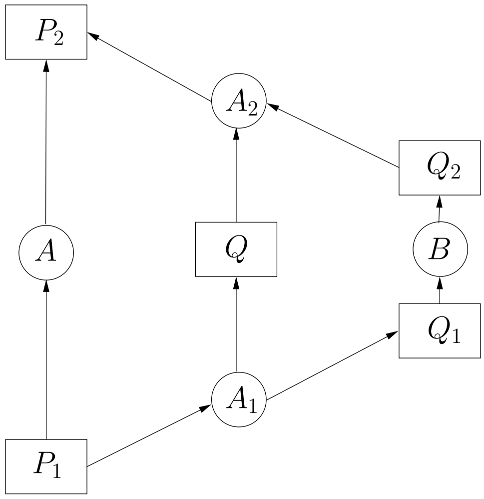

Figure 2: Hierarchical refinement.

In general, this set H must always form a hierarchy, i.e., should have the form of a tree, with the root account at the top. In our present example, H is just a flat hierarchy; in the next example we will see a non-flat hierarchy.

Now continuing our example, we see that we can switch the bottom and the top refinements on and off independently:

- To view only the basic vertical chain on the left, we take the union of the account views α_{root}, α_A^-, and α_Q^-.
- To view the bottom refinement but not the top, we take the union of α_{root}, α_A^+, and α_Q^-.
- Similarly, to view the top refinement but not the bottom, we take α_{root}, α_A^-, and α_Q^+.
- Finally, to view full detail, we take α_{root}, α_A^+, and α_Q^+.

In a good design, all these unions must yield legal OPM graphs even if we would erase all the accounts.

2 Hierarchical refinement

An example of an OPM graph with a refinement hierarchy of two levels is shown in Figure 2. The idea again is that the vertical chain $P_1 \rightarrow A \rightarrow P_2$ describes the overall process. The middle subgraph $A_1 \rightarrow Q \rightarrow A_2$ gives more detail about artifact A; moreover, the right subgraph $Q_1 \rightarrow B \rightarrow Q_2$ gives further detail about process Q.

Generalizing what we did in the first example to two levels, we introduce the following refinement accounts:

- A root account α_{root} covering the two nodes P_1 and P_2.
- Refinee account α_A^- covering the left vertical chain $P_1 \to A \to P_2$, and refiner account α_A^+ covering the edges $P_1 \to A_1$ and $A_2 \to P_2$.
- Refinee account α_Q^- covering the chain $A_1 \to Q \to A_2$, and refiner account α_Q^+ covering the chain $A_1 \to Q_1 \to B \to Q_2 \to A_2$.

Note that α_A^+ does not contain Q, as this node is subject to further refinement; the node is covered by α_Q^- instead.

The hierarchy H now consists of the pairs $(\alpha_{\text{root}}, \alpha_A^-)$ and (α_A^-, α_Q^-); note that it is not flat but has depth two. Note also that in our methodology, H always consists only of root account and refinee accounts, as it are the refinees that are either retained in a view or replaced by their refiner. Moreover, *a refinee can be refined only if its parent has been refined.* This is a new aspect that we did not see in the first example, which was flat.

Thus, in our example, we have the following unions of views:

- The basic view consisting of α_{root} and α_A^-;
- The first-level view consisting of α_{root}, α_A^+, and α_Q^-;
- The second-level view consisting of α_{root}, α_A^+, and α_Q^+.

Note that all these unions yield legal OPM graphs even when erasing all the accounts.

The general methodology (which we have followed here for α_A^-) is that for a refinee account α^- that is not a leaf, i.e., that has further children down the hierarchy, the account α^- contains the high-level view, whereas the account α^+ contains that part of the detailed view that is not involved in further refinements. So, α^+ plays the role of a root account, but at a deeper level.

3 General theory

In general we use a set *RefAcc* of accounts that are used to indicate refinements; we call these the *refinement accounts.* We will impose a structure on *RefAcc* consisting of the following:

- a distinguished root account α_{root};
- two disjoint sets *Refinee* and *Refiner*, both not containing the root;
- a bijection *Refine* from *Refinee* to *Refiner*;

- A hierarchy H on $\{\alpha_{\text{root}}\} \cup \mathit{Refinee}$ in the form of a rooted tree, the root of which is the root account.

We refer to the whole structure as a *multiple hierarchical refinement.*

It is convenient to denote refinee accounts as α^- and to denote $\mathit{Refine}(\alpha^-)$ as α^+. Also, for any account α in H, we will denote its parent by $H(\alpha)$.

If a refinee account α^- is not a leaf in H, i.e., has further children down the hierarchy, its refiner account α^+ must contain only that part of the refinement of α^- that is not involved in further refinements. If α^- is a leaf, then α^+ simply contains the whole refinement. We have seen this in the examples above.

Now we describe the general mechanism by which refinements can be switched on and off. We will see that we can traverse a space of configurations in which some refinements are made and others are not. Configurations will be formalized by "upward-closed" sets of refinee accounts, with the addition of the root account. Initially, we start in the situation where no refinements are made. This corresponds to the initial configuration $C_{\text{root}} = \{\alpha_{\text{root}}\}$. In general, from any reachable configuration C, we can make a number of possible "switch on" and "switch off" moves:

- If $\alpha^- \in C$ and β^- is a child of α^- not yet in C, we can move to the configuration $C \cup \{\beta^-\}$. We denote this new configuration by $\text{on}(C, \beta^-)$.
- If $\alpha^- \in C$ such that no child of α^- is in C, we can move to the configuration $C \setminus \{\alpha^-\}$. We denote this new configuration by $\text{off}(C, \alpha^-)$.

Note that the configurations that can be reached from C_{root} by any sequence of such moves are precisely all the upward-closed sets: sets with the property that if an element belongs to the set, then its parent also belongs to the set.

Now, for any such configuration C, we define the corresponding *view set* of C, denoted by $\overline{C}$, as follows:

$$\overline{C} = \{\alpha_{\text{root}}\} \cup \left\{\alpha^+ \mid \alpha^- \in C\right\} \cup \left\{\beta^- \mid \beta^- \notin C \text{ and } H(\beta^-) \in C\right\} .$$

That is, the view set of C always contains the root account, the refiner accounts of all the refinee accounts in C, and, the refinee accounts of the children that are not in C.

This whole hierarchical refinement system is called *legal* if for every configuration C that is reachable from C_{root}, the union of views from the view set $\overline{C}$ yields a legal OPM graph, even if we were to erase all accounts.