

3.1 - Use Case Leveling Approach

A use case refinement approach proposed by Cockburn and tailored for the purpose of the NCI was applied next (step 4). In this approach, use cases are structured at five levels, each identified in a row of Table 3.1-1 below. The "Community detail" and "UML diagrams" columns introduce RM-ODP community concepts and associated UML notation. The content of these columns will be discussed further in subsequent text.

Level	Use case detail	Community detail	UML diagrams
Cloud (level 0)	Summary Goal States high level business process (e.g. treat a patient for cancer)	Community objective (User focus) Depending on the business problem, may or may not identify system role May add some policy constraints	UML4ODP: Community contract; <ul style="list-style-type: none"> User role; Community Objective Policy package
Kite (level 1)	Summary Goal Further level of process detail, e.g. diagnostic or treatment specific. (e.g. treat a patient for cancer by using diagnostics, education, treatments, etc)	Sub-community sub-objective (if any) in complex communities; Depending on the business problem, might or might not identify system role; May add policy constraints	UML4ODP: Community contract (from above) plus <ul style="list-style-type: none"> Sub-communities (each specified with its community contract) Relationships between sub-communities
Sea (level 2)	User Goal Level Something the actor is trying to get done (e.g. treat a patient for cancer by ordering these lab tests, evaluating the results, customizing a treatment to specifically address concerns etc) NB: level 2 must drive out the major model aspects (static /information model, behavioral model, governance model, etc.). Can include some exception conditions.	High-level interaction, between User and System; identifies User and System roles explicitly; identifies artifact(s) involved	UML4ODP: <ul style="list-style-type: none"> Community contract (from level above); System Role Interaction (high-level, e.g. initiator /responder) between User and System Role Artifacts (information object) accessed, used or exchanged in interaction Identify policies that affect behavior of the roles and interactions in community
Underwater (level 3)	Sub-function Goal - needed to accomplish a user goal. Use cases at this level can typically be used and reused	As above but detailed interaction between User and System Roles, in terms of steps /sequence of steps involved	UML4ODP: <ul style="list-style-type: none"> the same as in level 2 above PLUS UML sequence (or activity) diagrams providing detailed steps in process - leading towards interface and service specification
Clam (level 4)	Sub-function Goal - not use cases per se - rather services /functions that implement	Each interaction supported by service interface; possibly use business rules derived from policy to constrain the behavior of roles	Define services as identified above; consider use of SAIF behavioral framework (BF) or other approaches, e.g. SoaML

Table 3.1-1. Use Case levels and Related Architecture Artifacts

These different use case levels are valuable as a way of presenting requirements to various stakeholders and end-users, depending on their expertise and responsibility. In a way, this classification through levels provides a complement to the existing classification of use cases using stories. Note that the existing stories could mostly be categorized as Sea level use cases.

Further, these levels have been implemented as UML stereotypes, allowing their classification in the Sparx EA tool as well as showing their traceability to the requirements documented on the wiki pages for the project. By applying this method, business analysts involved in the effort have modelled most of the use cases in the EA tool and these models will be published on the wiki pages.

An **example** of Use Case levels and related architecture artifacts follows. For illustrative purposes in this and the remaining sections, we will consider the 21090 Datatype Support requirement described on the page [Init1pm16.pm4 - 21090 Datatype Support](#). This use case can be considered a Kite level use case, because it describes new functionality of the NCI semantic infrastructure to support 21090 datatype management for Information Modelers.

This use case can then be further refined into a set of Sea level use cases, namely:

- Import Datatypes
- Localize Datatypes
- Generate XML Mapping
- Register Datatypes
- Interoperate on Datatypes
- Translation Services

Note that other classifications may also be possible and this is somewhat dependent on the modeller's choice. For example, with a greater level of detail these use cases could also be considered Underwater use cases.