Granularity in Conceptual Modelling: Application to Metamodels

Brian Henderson-Sellers¹ and Cesar Gonzalez-Perez²

¹ Faculty of Engineering and Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW 2007, Australia
² The Heritage Laboratory (LaPa) Spanish National Research Council (CSIC) Santiago de Compostela, Spain
Brian.Henderson-Sellers@uts.edu.au, cesar.gonzalez-perez@iegps.csic.es

Abstract. The granularity of conceptual models depends at least in part on the granularity of their underpinning metamodel. Here we investigate the theory of granularity as it can be applied to conceptual modelling and, especially, metamodelling for information systems development methodologies. With a background context of situational method engineering, this paper applies some theoretical works on granularity to the study of current metamodelling approaches. It also establishes some granularity-related best practices to take into account when adopting a metamodel, especially for its future use in developing method fragments for situational method engineering. Using these best practices will result in components of better quality and, consequently, better conceptual models and methodologies.

Keywords: concepts, granularity, metamodels, abstraction, modeling.

1 Introduction

In conceptual modelling we represent concepts with abstractions variously named “entity”, “class”, “type”, “object”, “agent” etc. (called here simply “entity” for generality). Although these entities (should) follow a definition in terms of either an ontological definition or specification by means of a metamodel (itself a conceptual model), there is no guarantee that the “size” of the resulting entities will be consistent across the model (be it metamodel or conceptual model).

In the context of conceptual modelling at both these levels (metamodel as well as model), we identify from the theoretical literature an underpinning theory for abstraction. The theory is then available to assist us in, firstly, determining a good quality set of meta-elements that will not only be self-consistent but that will, in due course, be able to generate “instances” (in the conceptual model) that are themselves of a consistent “size”. An example of where this is relevant is in the area of situational method engineering (SME) [1] [2] [3] in which each conceptual “instance”, known as a
method fragment, is used as the basic “building block” for the creation and ultimate use of a software development methodology.

In this paper, we therefore investigate how to apply a theory of granularity abstraction (Section 2) to both the metamodel and conceptual model level particularly in the context of software development methodologies. Following this discussion, in Section 3 we compare and contrast a number of methodology metamodels, focussing on work units, and evaluate how these are modelled in a variety of standard process-focussed metamodels (considered as conceptual models). In Section 4, we identify some guidelines for the best practice application of granularity in conceptual modelling. Section 5 outlines some related work before concluding in Section 6.

2 Granularity Abstraction

Abstraction refers to a process by which detail is removed in order to simplify our understanding of some entity or system (e.g. [4]). During such removal via a mapping between two representations, it is necessary to retain the essence of the system such that the simplification can be used effectively as a surrogate for the more detailed (and thus less abstract) entity/system. The aim of this process is to provide assistance with the original problem (e.g. [5]).

Formally, an abstraction can be defined as a mapping between two languages which may be similar or dissimilar (e.g. [6]) and defined as [5]:

\[ f : \Sigma_1 \Rightarrow \Sigma_2 \text{ is a pair of formal systems } (\Sigma_1, \Sigma_2) \text{ with languages } \Lambda_1 \text{ and } \Lambda_2 \text{ respectively and an effective total function } f_{\Lambda_1} : \Lambda_1 \rightarrow \Lambda_2 \]  

(1a)

or more simply [4] as

\[ \text{given two languages } L_0 \text{ and } L_1, \text{ abs } : L_0 \rightarrow L_1 \text{ is an abstraction.} \]  

(1b)

where abs is an abstraction function, \( L_0 \) is known as the ground language and \( L_1 \) the abstract language.

Having specified what an abstraction is, we can now look at the particularities of the granularity abstraction. By granularity, we mean (loosely) the number of entities in different representations of the same conceptual model. We can thus talk about a model that consists of a small number of entities as having coarse granularity or of a large number of entities as possessing fine granularity [7].

Hobbs [8] (quoted in [6]) argues that when we conceptualize the world, we do so at different levels of granularity, these levels being determined by the number of characteristics that are seen as relevant to a particular purpose. For example, a forest is composed of trees; that means that the forest (the composite) can be used as a granular abstraction of the trees in it (the components). Granularity relationships can be taken to the concept plane: the forest concept can be said to be a granular abstraction of the tree concept.

Lakoff [9] also deals with granularity in terms of the gestalt effect: he says that our brains and psychology, as human beings, are tuned to the perception of entities in a quite narrow range of spatial and temporal scales. For example, we do not immediately