A Proposal for Folding in ORM Diagrams

Tony Morgan

INTI International University, Malaysia
goldbasilisk@live.com

Abstract. Object Role Modeling (ORM) can be used to produce conceptual models of any scale. Although small examples are often used to illustrate specific points, it is equally possible to produce large models covering extensive business domains. To help users navigate around a large model, the ORM diagram is typically split over several virtual pages. This gives a fairly coarse granularity and it is difficult for users to change the page structure to meet some temporary viewing need. This paper describes an alternative approach that allows objects of lesser interest to be temporarily 'folded' away and 'unfolded' when required. The folding and unfolding can be dynamically controlled by the user, but operates under the control of specific algorithms. Some possible algorithms are discussed and illustrated with examples, and a number of usability issues are raised.

1 Introduction

Modeling is an occupation that shares many features with programming. Amongst these is the need for the developer of the model (or program) to interact with a representation of the model constructs (or code) during modeling (or programming). In particular a modeler (or programmer) often needs to focus on different parts of the model (or code), depending on the need at hand and in a way that cannot really be predicted in advance.

A program listing can be visualized as a long strip of paper down which appears successive lines of code: older programmers will remember the physical manifestation of code in earlier times as a 'line printer listing'. This structure has a principal organization that is one-dimensional. Moving to a region of interest in the listing is accomplished by moving 'up' or 'down' the conceptual strip. Once the region is located, 'left' and 'right' scanning can be employed, but the 'horizontal' dimension is clearly subsidiary to the 'vertical' dimension.

Programmers often need to juxtapose segments of code that might be widely separated in the code listing. For example, some code may be defined in one place but used other places that are not immediately adjacent. During activities such as debugging it is convenient to see the definition and usage segments together whilst hiding other segments that are not immediately relevant. Given the linear structure of the code, this can be accomplished by conceptually folding a listing so that the regions of interest are brought together. This is illustrated in Figure 1, where the folding has brought the regions shown as A and B together, whilst hiding the intervening region which is assumed to be of no interest.

R. Meersman et al. (Eds.): OTM 2010 Workshops, LNCS 6428, pp. 474–483, 2010.