

Differential Conversion: DWG – SVG Case Study

Martin Ota and Ivan Jelínek

Czech Technical University in Prague, Department of Computer Science and Engineering,
Praha 2, 131 25 Karlovo nám. 13, Czech Republic
{otam,jelinek}@fel.cvut.cz

Abstract. This paper is focused on differential conversion, which is a new approach to data transformation between different software systems. The strategy of differential conversion, which is built on a log of changes and identifier mapping, was introduced in CDVE 2006 in a purely theoretical paper entitled *Data Exchange in CAD during Iterative Work with Heterogeneous Systems*. A specific case study of differential conversion in use is elaborated in this text. It is about an intelligent bi-directional transformation between DWG and SVG. DWG is a proprietary CAD-format of the AutoCAD system, whereas SVG is an open, generic graphics format, based on XML. The implemented transformation allows usage of a model created in a CAD-system within a cooperative design intended for non-CAD users. It is useful not only for reading, but even for writing into the model and modifying it. Any changes made are returned back to the original model. The following case exemplifies the previously discussed strategy. For example, a project submitter, or another party in the review process, can use a normal, generic SVG editor for writing comments and proposals into the model. A solver can transform them back into the CAD-model, without other entities in the model being damaged. This case study describes differential export and import tools and takes note of the particular solutions of the log of changes and identifier mapping problems. It also describes the relationship to the original differential conversion proposal, discusses the particular implementation, and again generalizes into some *design pattern* for differential conversion implementation projects in the conclusion.

Keywords: CAD, cooperative design, cooperative engineering, data exchange, differential conversion, SVG.

1 Introduction

Differential conversion (DC) was introduced in CDVE 2006 [1] as a way of exchanging data during cooperative engineering and design between heterogeneous systems, which minimizes data distortion during its conversion. Utilization of DC is meaningful when the data is converted for modification in another system and then converted back to the original format. The main idea of DC is to transfer only new and changed objects and the objects without any changes are kept from the original data representation. The generic proposal of DC uses identifier mapping and the log of changes to reach it. The identifier mapping controls the preservation of information, which object was transformed into what (it cannot be simply a 1:1 relation), and the log of changes,

which must be supported by the system when data is modified after the conversion and is used for identifying changes in objects in the temporary data format. *New*, *modified* and *deleted* states are distinguished. The state of the object that is not in the log is *no change*.

Since our research is focused on CAD, we therefore deal with DC in this area as well. The following text describes a specific implementation of DC between two considerably different graphic formats – DWG and SVG. The domain of the case study is the design of the interior of an office building.

2 Case Description

The requirements are the following: The output of the office building design is, among others, the interior plan with a unified draft of the offices. Department managers should fill in comments and can eventually reconfigure the furniture, change the number of chairs and tables or change how the cubicle installations are divided. In this draft, they should fill in the names of employees on the single chairs. Managers modify only that part of the interior where their department is going to be placed. Merged modifications should be incorporated into the final model, which should be published for the employees on the intranet in two stages. First, the anonymous draft, and afterwards should come the modified final model with the employee names.

The building model is in DWG format of AutoCAD 2007. The managers do not have this system and thus is economically required to use a freeware editor based on SVG [2]. SVG is planned to be used for the intranet presentation of the model.

The SVG model is equipped with instructions on how to edit it. For example, an employee name is connected to a chair as a prepared text element `EMPLOYEE_NAME`, which correspond with a block attribute in DWG, and should be modified by filling in the name. The next example is that adding furniture must be done as object copying, etc. The exact rules are not important for this description of the case study.

3 Case Solution

A solution of the case uses DC, but it is not possible to follow the basic form because the selected SVG editor does not allow adding the functionality for on-line log of changes creating. The SVG editor used has two qualities that subsequently helps deduce the log of changes. The first quality is that the SVG editor respects identifiers stored in the ‘id’ objects attributes. The second quality is that the representation of unchanged objects is kept identical, even after saving the SVG representation, which is changed in the other positions. Thus, the objects’ changes can be deduced from the comparison of the XML trees of the SVG representations before and after editing.

3.1 Process Model

Figure 1 depicts a conceptual process model of the case solution. The model does not contain a requirement solution for publishing on intranet, because it can be reached