

Leveraging Single-User Microsoft Visio for Multi-user Real-Time Collaboration

Kai Lin¹, David Chen¹, Chengzheng Sun², and Geoff Dromey¹

¹ School of Information and Communication Technology,
Griffith University, Brisbane, QLD 4111, Australia
{K.Lin,D.Chen,G.Dromey}@griffith.edu.au

² School of Computer Engineering, Nanyang Technological University, Singapore, 639798
CZSun@ntu.edu.sg

Abstract. Microsoft Visio is one of the most prevalent commercial single-user graphic editing systems, which can be used to create a wide variety of business and technical drawings. It is desirable to leverage single-user Visio system for multi-user real-time collaboration. One feature that distinguishes Visio from other graphic editing systems is that *formulas*, a type of constraint, are defined in Visio to express the attributes of each graphic object, and the relationship between different Visio graphic objects. The ability to describe shapes with constraints opens many possibilities for making shapes behave in complex and sophisticated ways, but satisfying constraints in the presence of concurrency in collaborative systems is a challenge. In this article, we introduce a collaborative Visio system, called CoVisio, which enables a group of users to view and edit the same Visio documents at the same time from different collaborating sites. The methods applied to develop CoVisio are generic and can be adopted to leverage other single-user systems that support constraints, such as Microsoft Excel, for multi-user collaboration.

Keywords: Collaborative Visio, Consistency maintenance, Constraint satisfaction.

1 Introduction

With the increasing importance of using computers to support collaborative work, it is natural to expect existing single-user computer applications to play an important role in supporting collaboration. Leveraging single-user commercial systems for real-time multi-user collaboration has been a popular research topic for many years. So far, pioneer researchers have successfully enriched Microsoft Word and PowerPoint with collaborative functions, without modifying the source code of MS Word and PowerPoint [6], [9]. This is achieved through the use of MS Word/PowerPoint's API (Application Programming Interface), to combine Word/PowerPoint with collaboration features.

Microsoft Word and PowerPoint represent two types of well-known single-user computer applications: text editing and presentation. Another type of popular computer applications is graphic editing that provides both generic and comprehensive graphic manipulation functions. Microsoft Visio is one of the most prevalent

commercial single-user graphic editing systems, which can be used to create a wide variety of business and technical drawings. It is desirable to furnish single-user Visio system with multi-user collaborative functions, so that users can work collaboratively in groups to improve productivity.

One feature that distinguishes Visio from other graphic editing systems is that *formulas* are defined in Visio [4]. Visio *formulas* are constraints that determine how Visio shapes look and behave on a drawing page. Visio users can precisely control the appearance and behavior of Visio shapes by editing *formulas*. The ability to describe shapes with constraints opens many possibilities for making shapes behave in complex and sophisticated ways, but satisfying constraints in the presence of concurrency in collaborative systems is a challenge [3].

This paper presents a collaborative Visio system, called CoVisio, which extends single-user Microsoft Visio for multi-user collaboration based on Visio API, so that a group of users can view and edit the same Visio documents at the same time from different sites. Both constraints and consistency are maintained in CoVisio.

The rest of this article is organized as follows. Section 2 introduces Visio *formulas*, which distinguish Visio from other graphic editing systems. CoVisio system is described in section 3, including CoVisio components, architecture and how to maintain both constraints and consistency in CoVisio. The major contributions and future work of our research are summarized in the last section.

2 Visio Formulas

An attribute of a graphic object, called a *cell* in Visio, is expressed by a *formula*. A *formula* is an expression that can contain constants, operators, and object attribute references. Microsoft Visio evaluates a *formula* to a result and then converts the result to the appropriate units for the attribute that contains the *formula* [4]. Some *formulas* consist of a single constant, but all *formulas* go through this evaluation and conversion process. In a Visio *ShapeSheet* window, a user can display *cell* contents as either *values* or *formulas* by clicking the appropriate command on the **View** menu.

Visio *formulas* may define the relationship between graphic attributes of a single graphic object. For example, the X and Y coordinates of the top-right vertex of a rectangle are defined by two *formulas*, *width*×1 and *height*×1, respectively. Therefore, each time a user resizes the rectangle (i.e. changes the *width* or *height* of the object), the *formulas* defining the coordinates of the vertex will be reevaluated, so that the position of the vertex will be changed accordingly.

On the other hand, a Visio *formula* may define the relationship between graphic attributes of different objects. For example, when a user connects the *begin point* of a line to a *connection vertex* of a rectangle, the *formula* defining *begin point* of the line is automatically changed to something like *PAR(PNT(Rectangle! Connections.X1, Rectangle!Connections.Y1))*, which means the *begin point* of the line should have the same coordinate as the *connection vertex* of the rectangle. Therefore, even if a user moves or resizes the rectangle, the *begin point* of the line still glues to the *connection vertex* of the rectangle.

It is worth to point out that users could not change the value of an attribute directly in Visio, as a value is always evaluated from a *formula*. Each time a user updates the