Expert Systems with Applications 38 (2011) 1089–1098

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Collaborative design: Improving efficiency by concurrent execution of Boolean tasks

Yang Zheng [a,*], Haifeng Shen [b], Chengzheng Sun [a]

[a] School of Computer Engineering, Nanyang Technological University, Singapore
[b] School of Computer Science, Engineering and Mathematics, Flinders University, Australia

ARTICLE INFO

Keywords:
Concurrent execution
Collaborative design
Design efficiency
Computer-Aided Design
Boolean operation
CSCWD

ABSTRACT

Computer-Aided Design (CAD) applications provide design and engineering professionals with various computer-based tools to perform design activities. As efficiency is one of the most important requirements in most design tasks, in this article, we contribute a novel collaborative design approach to improving efficiency, where a complex design task can be divided and executed concurrently by multiple collaborative designers. This approach is particularly effective for design tasks where Boolean operations – widely supported by most CAD applications – are heavily used in design activities, such as architecture design, mechanical design and digital media design. We have designed and implemented a prototype system CoAutoCAD to test the approach and to demonstrate a variety of collaborative design activities.

1. Introduction

Computer-Aided Design (CAD) applications provide design and engineering professionals with various computer-based tools to perform design activities. A CAD application can be used to design, develop and optimize products, which can be goods used by end consumers or intermediate goods used by other products. It is also extensively used in domains such as machinery design, manufacturing and urban planning, from small residential types (e.g., houses) to large commercial and industrial structures (e.g., hospitals and factories). Ever since introduced to its applicable fields, CAD applications have constantly been improved to meet specific requirements such as lowering product development costs, shortening design cycles and consolidating valuable ideas from various sources.

According to Gero (2000), design activities can mainly be classified into two kinds: routine designing and nonroutine designing. Routine designing produces designs that are some minor variations of existing ones, which means that most of the design tasks and details are known even before a design process starts.

Efficiency is an important requirement in most routine design tasks and various approaches have been developed to help improve design efficiency. One of the approaches is "collaborative design" or "collaborative engineering" (Shen, Hao, & Li, 2008), which allows designers from multidisciplinary domains and geographically dispersed locations to work on a common project. The approach is able to help coordinate routine design activities and share design information in a complex design project that involves multiple designers by employing advanced collaboration technologies, e.g., video conferencing (Egido, 1988) and application sharing (Begole, Rosson, & Shaffer, 1999).

In the past decade, we have witnessed the invention of lots of applications and tools in the fields of engineering, CAD and CSCWD to support routine designing. Those applications and tools are able to construct effective communication channels between distributed designers. They are also equipped with methods and techniques to support design planning (Li, Zhang, Gao, Li, & Shao, 2010; Yu & Li, 2006), which can divide a big design project into smaller design tasks, assign them to individual design teams/experts, and coordinate the integration process of the tasks.

While those methods and techniques are effective in improving design efficiency at a higher level (i.e., project level), not much work has been done to support improving efficiency at the lower level (i.e., task level). As today's design projects are usually huge and interdisciplinary, a design task divided from a project could still be complex enough for a designer to perform. As a result, the process of completing individual tasks is usually tedious and error-prone, and easily among one of the root causes that lead to design inefficiency.

To tackle this problem, in this article we propose a collaborative design approach to improve efficiency in routine design by further dividing a task into subtasks and allowing multiple designers to complete the subtasks in parallel. In practice, it is non-trivial to decompose a complex design task and integrate the subtasks. An expert system with collaborative functionalities must exist to help designers complete this process. Such a system should have the following features:

(1) *Division and integration*. A complex design task usually needs to be performed by multiple designers who have different

* Corresponding author.
E-mail addresses: zhen0042@ntu.edu.sg (Y. Zheng), hfshen@csem.flinders.edu.au (H. Shen), CZSun@ntu.edu.sg (C. Sun).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.05.004

specialties, habits and culture backgrounds. They can contribute to a design task from different perspectives. There are usually three steps in the process (which can be done in sequence or in parallel): dividing a task to multiple subtasks, allowing individual designers to perform the subtasks, and integrating the subtasks. While designers can use the functionalities provided by the design tools to perform the subtasks, the system should provide flexible mechanisms to help integrate the subtasks.

(2) *Tools and design environments*. Designers' proficiency in using a design tool is a key factor to determine the efficiency of completing a design task. A new tool should retain the "look-and-feel" and functionalities of the tools that designers are familiar with. Otherwise, design efficiency could be compromised by the learning curve.

(3) *Visualization and review*. In a collaborative design environment, a designer needs to be informed of others' work in order to avoid potential conflicts. This implies that the collaborative design tool should enable visualization of individual designers' work in the shared workspace in real time.

Therefore, we have designed and implemented a collaborative design tool CoAutoCAD (Zheng, Shen, & Sun, 2009), which integrates advanced collaborative functionalities into one of the most widely-used commercial-off-the-shelf design tool AutoCAD (Autodesk, 2010). CoAutoCAD retains AutoCAD's "look-and-feel" and conventional single-user features, but is equipped with advanced collaborative functionalities such as concurrent work, conflict resolution, and consistency maintenance.

In this article, we present a novel solution to improving efficiency in routine design, where a complex design task can be divided and executed in parallel by multiple collaborative designers. This solution is particularly effective for design tasks where Boolean operations – widely supported by most CAD applications – are heavily used in design activities, such as architecture design, mechanical design and digital media design. This solution has been implemented in CoAutoCAD for testing its effectiveness and demonstrating a variety of collaborative design activities where design efficiency can be significantly improved.

The rest of this article is organized as follows. In Section 2, we briefly review the design of the CoAutoCAD system. The technical background of tackling the negative side of conflicting operations in CAD applications is introduced in Section 3. Section 4 describes how to improve design efficiency by concurrent execution of Boolean tasks using CoAutoCAD. Finally, major contributions of this article and the future work are summarized in Section 5.

2. The design of the CoAutoCAD system

Collaboration has been increasingly needed in the CAD community and various CSCW (Computer-Supported Cooperative Work) technologies have thus been adopted to support the development of collaborative design systems. Existing collaborative CAD applications generally have the problems of not allowing concurrent work and poor local response. To tackle these problems, we convert the commercial-off-the-shelf single-user AutoCAD application to CoAutoCAD by the Transparent Adaptation (TA) technology (Sun et al., 2006).

2.1. CoAutoCAD architecture

By employing the TA technology, AutoCAD was transparently converted to CoAutoCAD without having its source code changed. For existing AutoCAD users, not only have the "look-and-feel" and functionalities of the AutoCAD application been preserved, but advanced collaborative functionalities have also been added. As shown in Fig. 1, AutoCAD is extended with advanced collaborative functionalities by the AutoCAD Collaboration Adaptor (CA), via which the Generic Collaboration Engine (GCE) is plugged into AutoCAD by means of their Application Programming Interfaces (API). GCE is a re-usable software component with application independent collaboration capabilities such as consistency maintenance, concurrency control, group awareness, interaction control, and has been successfully used to convert Office productivity applications Word and PowerPoint to CoWord and CoPowerPoint, respectively.

The CoAutoCAD system provides collaborative designers with an "unconstrained collaborative environment" – designers have the total freedom of modifying any data object in a shared design model at any time. This "unconstrainedness" is underpinned by the cornerstone Operational Transformation (OT) (Ellis & Gibbs, 1989) technology used by GCE component for optimistic concurrency control.

2.2. CoAutoCAD data adaptation

The TA approach requires data and operations of the AutoCAD application be adapted to those used by GCE. The data adaptation is based on the eXtended OT Data Model (i.e., XOTDM) (Sun et al., 2006), which consists of a hierarchy of addressing groups and each group consists of multiple independent linear addressing domains. Within an addressing group, independent linear addressing domains are identified by their unique names within that group. A data object is mapped to a position in a linear addressing domain only if it has the position number as its address in this domain. As a result, to address any data object in an XOTDM, a vector of (*n*,*p*) pairs is needed: $vp = [(n_0,p_0),(n_1,p_1),\ldots,(n_i,p_i),\ldots,(n_k,p_k)]$, where $vp[i] = (n_i,p_i)$ $(0 \leqslant i \leqslant k)$ represents one addressing point at level *i*.

Fig. 2 shows the data addressing model for CoAutoCAD, which is a two-level XOTDM. The top-level has four independent linear addressing domains, including *Model Spaces*, *Layers*, *Paper Spaces*

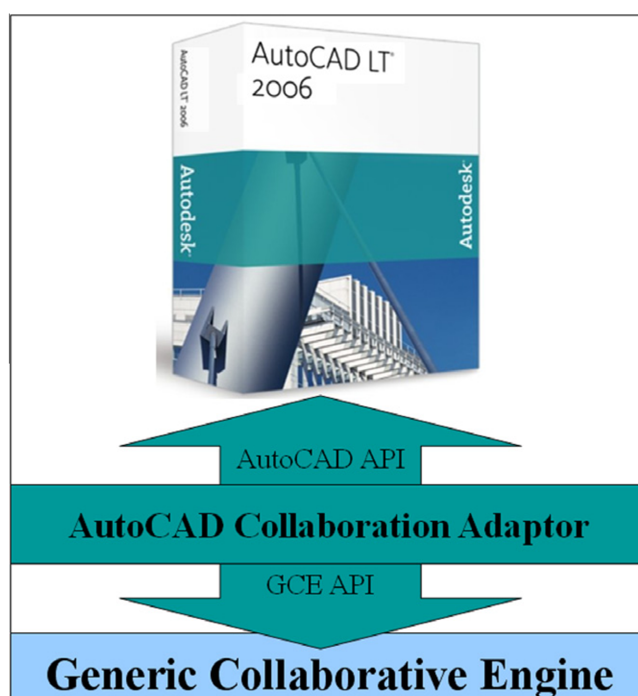

Fig. 1. CoAutoCAD architecture.