1. Introduction

Increasing emphasis is being placed on the information management function in the competition between modern businesses. In order to compete successfully in global markets, businesses must achieve excellence in managing their information management operations. Business information system (BIS) is seen as one of the key strategies that businesses should adopt in their efforts to achieve information management excellence. It comes as no surprise, therefore, that the academic and industrial community continually attempt to develop technologies that will make it easier, faster, and less expensive to build and maintain high-quality BISs. ActiveX component technology (Chappell, 1996; Gray et al., 1998; Hoque & Sharma, 1998; Microsoft, 2007b; Oreizy & Kaiser, 1997) is an excellent example of these technologies. At present, ActiveX component technology has been widely applied in system software and tool programs, such as Adobe's Flash player, Microsoft's online update service, and Trend Micro's HouseCall online antivirus program. Unfortunately, the relative success in system software and tool programs has not been matched by the same degree of success in BISs. However, the ActiveX component technology does, in fact, have the potential to contribute to the attainment of information management excellence in businesses, and therefore

possible new applications of ActiveX component technology in BISs are worthy of continued study.

At the same time, remarkable advances in personal computing technologies have drastically affected the building of BISs. There has been a shift in BISs from using a massive mainframe computer, centralized architecture, and text-based user interface to a personal computing device, client/server (C/S) architecture, and graphical user interface. Not very long ago, large-scale mission-critical BISs were the exclusive province of massive mainframe computers. That is changing rapidly. Today, increasingly large and complex BISs are being built as C/S applications. Unfortunately, for the most common C/S BISs, that is, the Windows-based C/S BISs (WinBISs), client programs must be manually deployed to each end-user machine bringing about a heavy BIS maintenance load. ActiveX component technology, if used properly, is more cost-effective than other alternatives in improving the deployment of existing WinBISs. That is to say, ActiveX component technology will enable developers to leverage their existing knowledge, skill, tools, experience, and code set to automate the deployment of existing WinBISs, meaning costly and risky rewrites of existing WinBISs can be avoided. Nevertheless, ActiveX component technology is no magic solution, and thus still has three shortcomings, namely insecurity, complexity, and bulkiness. Presumably, these

shortcomings resulted in fewer applications and less research of ActiveX component technology in BISs.

The purpose of the study presented in this paper is to overcome the shortcomings of ActiveX component technology, and then develop an ActiveX component-based solution to automate the deployment of existing WinBISs. The following products are obtained from the study:

1. The answers to the shortcomings of ActiveX component technology.
2. The first part of the proposed solution, namely the downloadable architecture, which supports the development of automatically-deployed BISs.
3. The second part of the proposed solution, namely the architecture transformation process, which transforms existing WinBISs into the downloadable architecture.
4. Two examples of how to use the downloadable architecture and the architecture transformation process. The examples also provide evidence to support the feasibility of such architecture and process.

The remainder of this paper is organized as follows. Chapter 2 is an extensive review of the literature. Chapter 3 describes the methodology and procedures to be used to accomplish the specific purpose of the study presented in this paper. Chapter 4 discusses the shortcomings of ActiveX

component technolog and their answers. Chapter 5 clarifies the assumptions of the proposed solution. Chapter 6 describes the downloadable architecture. Chapter 7 describes the architecture transformation process. Chapter 8 and 9 show two examples of how to use the downloadable architecture and the architecture transformation process, while chapter 10 concludes the paper.