The purpose of this research is to explore how international companies adopt and enforce standards in information systems and technology. Standardization of hardware, software, and data is often perceived to be beneficial for an organization. However, for multinational companies, differences among nations in language and culture, law and regulation, and business practices and climate, among other factors, combine to thwart standardization efforts. This research examines the problems that a number of multinational companies have faced and the solutions they have pursued in attempting to standardize their systems. The lessons learned from these case studies offer a prescription for improving the standardization process.

The absence of hardware, software, and data standards is widely recognized as a major impediment to the development of global information systems (Alavi & Young, 1992; Barr et al., 1988; Palframan, 1991). The trade and academic literature has focused on standards that governments, vendors, and industry groups are struggling to adopt. By contrast, little research has addressed a related problem: even if such groups could agree on a consistent set of hardware and software standards worldwide, the absence of intra-organizational standards at many companies would continue to impede their ability to exchange and integrate data among their various, semi-autonomous information systems (Ives & Jarvenpaa, 1991; Karimi & Konsynski, 1991; McQuillan, 1989).

At many multinational companies, the absence of intra-organizational standards derives in part from the independent evolution of systems developed by subsidiaries in different countries at a time when the benefits of global integration were underappreciated and when global systems were technically difficult to implement (Roche, 1992). Recently, managers have increasingly recognized that global information systems are necessary to support global strategies (Alavi & Young, 1992; Deans & Kane, 1992; Keen, 1992). Concurrently, technical hurdles to global integration have decreased (Kanter & Kesner, 1992). Nevertheless, many companies find that their subsidiaries in different countries continue to use different software packages and to operate them on different and incompatible platforms (Roche, 1992).

While divisionalized domestic corporations share a similar history of software and hardware incompatibility, domestic companies are driven to standardize primarily by business needs and economics, moderated perhaps by organizational considerations. The decision for multinational companies (MNCs) is likely to be more complicated because of differences among countries in a variety of factors including language and culture, regulation and law, and business practices and environment (Deans & Ricks, 1991). The purpose of this paper is to explore how global companies have addressed the standardization problem and to learn from their suc-
cesses and failures what pitfalls to avoid and what approaches to pursue.

For purposes of this paper, an “organizational standard” (hereafter referred to as a “standard”) is defined as a set of rules or policies governing the characteristics of data, software, and/or hardware that an organization may purchase, develop, or maintain. This concept differs from that of an “industry standard,” a set of rules adopted by a majority or large plurality of software and hardware vendors in a particular industry governing the characteristics of the products that they sell. Following Irwin (1991), we also distinguish between organizational standards and policies. Policies are rules relating to the management, rather than characteristics, of software and/or hardware. For example, the requirement that a license be obtained for all software purchased from third parties is a policy, whereas the requirement that all software purchased from third parties run under the UNIX operating system is a standard.

This paper is organized as follows. First, it describes the background for this research and outlines a normative model to motivate and support the research direction. Next, it addresses the methodology and presents the collected data. It then discusses the findings and suggests some approaches for improving the standardization process. Finally, it draws conclusions and presents some hypotheses for further study.

Background and Motivation

Extensive academic research and practitioner evidence indicates that companies operating in multiple countries face a variety of issues and constraints in the management of information technologies that compa-

**BENEFITS**
- Minimizes duplication of software development
- Increases the quality of developed software
- Increases integration of systems for improved efficiency
- Increases ability to exchange data among systems
- Achieves economies of scale in purchasing and maintenance
- Improves negotiating position for better pricing
- Promotes and facilitates coherent mission and strategy
- Reduces applications development time and cost
- Reduces outside projects with run-away costs
- Increases flexibility in use of IS personnel
- Reduces the cost and increases the quality of support
- Reduces training costs and time
- Lifts burden of product research from the user
- Reduces number of specially-built interfaces

**DRAWBACKS**
- Reduces flexibility in applications
- Stifles innovation and creativity
- Interferes with other requirements of applications
- Reduces ability to go with lowest cost solution in each case
- Consumes political good will
- Increases frequency of revision and upgrade installation
- Decreases users comfort about opportunities to meet their direct needs
- Increases impact of any major changes
- Decreases ability to make major changes
- Requires more review and consensus for software/hardware selection
- Increases impact of poor decisions
- Increases costs of purchasing due to reduced supplier options
- Impedes the acquisition of new technology

---

Table 1: Benefits and Drawbacks of Standardization

<table>
<thead>
<tr>
<th>BENEFITS</th>
<th>DRAWBACKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimizes duplication of software development</td>
<td>Reduces flexibility in applications</td>
</tr>
<tr>
<td>Increases the quality of developed software</td>
<td>Stifles innovation and creativity</td>
</tr>
<tr>
<td>Increases integration of systems for improved efficiency</td>
<td>Interferes with other requirements of applications</td>
</tr>
<tr>
<td>Increases ability to exchange data among systems</td>
<td>Reduces ability to go with lowest cost solution in each case</td>
</tr>
<tr>
<td>Achieves economies of scale in purchasing and maintenance</td>
<td>Consumes political good will</td>
</tr>
<tr>
<td>Improves negotiating position for better pricing</td>
<td>Increases frequency of revision and upgrade installation</td>
</tr>
<tr>
<td>Promotes and facilitates coherent mission and strategy</td>
<td>Decreases users comfort about opportunities to meet their direct needs</td>
</tr>
<tr>
<td>Reduces applications development time and cost</td>
<td>Increases impact of any major changes</td>
</tr>
<tr>
<td>Reduces outside projects with run-away costs</td>
<td>Decreases ability to make major changes</td>
</tr>
<tr>
<td>Increases flexibility in use of IS personnel</td>
<td>Requires more review and consensus for software/hardware selection</td>
</tr>
<tr>
<td>Reduces the cost and increases the quality of support</td>
<td>Increases impact of poor decisions</td>
</tr>
<tr>
<td>Reduces training costs and time</td>
<td>Increases costs of purchasing due to reduced supplier options</td>
</tr>
<tr>
<td>Lifts burden of product research from the user</td>
<td>Impedes the acquisition of new technology</td>
</tr>
<tr>
<td>Reduces number of specially-built interfaces</td>
<td></td>
</tr>
</tbody>
</table>