A CASE FOR INFORMATION OWNERSHIP

IN ERP SYSTEMS TO ENHANCE SECURITY

Prof. S.H. von Solms, M.P. Hertenberger

Rand Afrikaans University, Johannesburg, South Africa

Prof. S.H. von Solms

Email address: basie@rau.ac.za

Mobile telephone number: +27 82 553 2436

Fax number: +27 (11) 489 2138

Postal address: PO Box 524, Auckland Park, 2006, South Africa

M.P. Hertenberger

Student number: 8914123

Email address: manfred.hertenberger@sbs.siemens.co.za

Mobile telephone number: +27 83 377 0921

Fax number: +27 (11) 652 7411

Postal address: P.O. Box 2838, Northriding, 2162, South Africa

ABSTRACT

This study investigates the lack of information ownership in current Enterprise Resource Planning (ERP) software systems. The purpose is to show how difficult, time consuming and costly the implementation of security within such systems is. The focus is on the investigation of security implementations within well-known ERP software packages such as SAP R/3 and Oracle E-Business Suite. The results of the study indicate that central administration, control and management of security within the ERP systems under investigation weaken security. It was concluded that central administration of security should be replaced by a model that distributes the responsibility for security to so-called information owners. Such individuals hold the responsibility for processes and profitability within an organization. Thus, they are best suited to decide who has access to their data and how their data may be used. Information ownership, coupled with tight controls can significantly enhance information security within an ERP system.

KEY WORDS

Database security; security policy; misuse detection; authentication; information flow.

A CASE FOR INFORMATION OWNERSHIP

IN ERP SYSTEMS TO ENHANCE SECURITY

1 INTRODUCTION

Enterprise Resource Planning software systems are in use by many different organizations and businesses worldwide. To facilitate the understanding of the reader, the term ERP is defined here as any software system that has been designed to support and automate the business processes of medium and large businesses.

Due to the fact that the ERP system in use by an organization contains critical business data, it is essential that such information be protected from unauthorized access. Unauthorized access to the data within the ERP system's database must be prevented. According to some sources, a large percentage of fraud takes place within the organization[1,10].

To protect all data within the ERP system's database, data security has traditionally been implemented by providing a centralized security infrastructure or subsystem within the ERP system. Such a security subsystem generally allows an administrator to define profiles and roles for each user accessing the system. Once the user has received a logon name and a password, the system permits access only to the areas permitted by the roles allocated to the user master record. If required, the user can be restricted to certain screen forms and be prevented from entering certain values in various fields. Though the centralized security subsystem is workable, certain problems can be identified. It is the view of the authors that security can be increased substantially by changing the centralized view on security to a decentralized one. This approach is briefly discussed in this paper.

2 STUDY RESULTS

The approach introduced above is implemented in most modern and current ERP software packages. During the course of this study, four different ERP software packages were investigated to determine the method used to implement a security subsystem. The following ERP software packages were investigated:

- SAP R/3 by SAP AG[2];
- Oracle E-Business Suite by Oracle Corporation[3];
- Navision Attain by Microsoft Corporation[4];
- Navision Axapta by Microsoft Corporation[5]

The above products were selected based on market penetration and target segment. This permitted various approaches and technologies to be compared. The security subsystem of each product was researched and its implementation and specific features critically examined. The aim of the product investigation was to determine the current state of the art of ERP security in products in everyday use.

The primary conclusion of the investigation into the security implementation and provision of the above products was that all rely on a traditional, centralized approach to information security. Though specific implementation differences do exist, most follow the route of requiring the creation of profiles, roles or permissions[6]. The profiles, roles and permissions determine various actions that may be completed within the system. Possible actions may be the generation of a report, the entry of a sales order or the execution of a program, for example. Once these actions have been translated

into profiles, roles or permissions, the security administrator assigns them to individual user master records. The user is presented with a user name and password. Once a logon action has been completed, the user is restricted to performing only those tasks that are permitted by the roles allocated to the user master record.

3 THE CENTRALIZED APPROACH TO SECURITY

The approach provided by the software products under investigation relies on one or more administrators to determine and define the security requirements centrally. In other words, one or more administrators are tasked with the creation and generation of roles and profiles that are allocated to user master records. Once these roles and profiles have been created, the user to whom they are allocated is able to complete only a certain subset of possible tasks within the system.

In contrast to legacy systems, ERP systems provide functionality specific to a user's requirements directly to that user's desktop. However, security is still configured centrally. This is a very traditional approach that mirrors administration in legacy software systems and environments. The results of the investigation into various ERP software packages concluded that the centralized approach to security is not ideal. Particularly, the practical implementation of security within ERP environments is often error-prone, time consuming and very costly when completed in the centralized fashion.

A diagrammatic representation of the traditional centralized approach to security within ERP environments is presented in Figure 1 below:

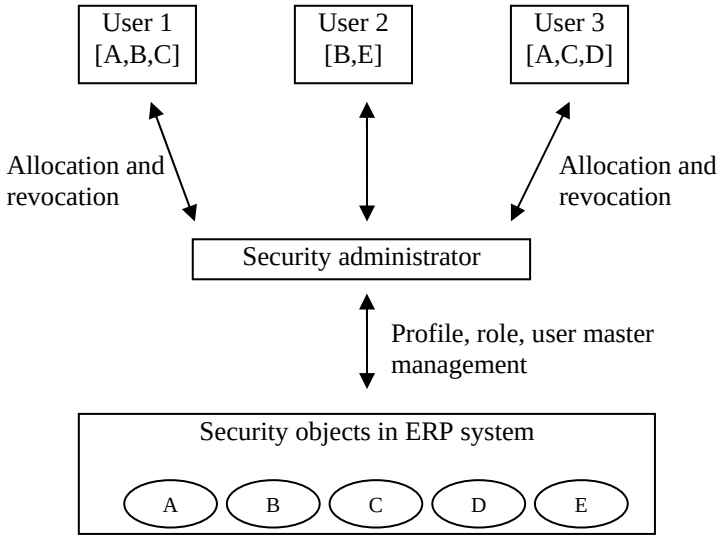

Figure 1. Centralized security within an ERP environment

A number of the problems associated with centralized security implementations are discussed below.

3.1 Error-prone configuration of the security subsystem

The central approach to a security implementation within an ERP environment is faced with numerous challenges. As an ERP software package supports all facets of the organization or business, numerous functional areas have to be covered by the software package. For example, a large manufacturing organization may require financial, sales and material management functionality. For each of these functional areas, very specific objects, data types and processes

have to be configured within the ERP software package, to support compliance with the organization's strategy and operation. As each such functional area is under the supervision of skilled and experienced employees and members of the organization, a very specific pool of knowledge is present.

To assume that a central system or security administrator has the ability to understand all nuances and specifics of each such functional area is often incorrect. Instead, the security administrator must gather information from each area of the business. Once all these details have been gathered, the security administrator is able to translate the requirements of each business area into the appropriate roles and profiles within the ERP system. In many cases, the security administrator has to select objects manually to create the appropriate access authorization for the user. It should be clear that such a process is often completed with a number of errors and omissions.

3.2 Time consuming and costly configuration of the security subsystem

ERP systems are generally installed in organizations with a large number of users. The creation of profiles and roles for a user population in excess of 200 users becomes a very complex and administration-intensive operation. As mentioned in the paragraph above, the central creation and maintenance of user master records, roles and profiles assumes knowledge of the specific functional area for which the security and access authorization settings are to be made.

Though all ERP systems that were investigated during the course of this study provide a means of creating profiles and roles in an accelerated fashion, very little support is provided to ensure accurate creation of security objects. This means that the security administrator, together with a knowledgeable member of the organization, has to spend a significant time testing the validity of the created access rights of each user. This testing cycle often involves an entire project team and is often completed during the testing of business process mapping within the ERP system. Unfortunately, testing is often performed with only a single user type in mind. This means that users with fewer access permissions may be denied access to certain functions that are required by them. In other cases, users holding more access authorizations may be able to access parts of the system that should not be accessible to them. Due to the fact that the testing process requires manual intervention, the requirement for additional project resources often increases costs. If incorrect decisions were made during the creation of access authorizations within the system, substantial amounts of work may have to be repeated.

3.3 Lack of change management and documentation support

The points made in the paragraphs above involve time, cost and knowledge constraints that are apparent due to the centralized nature of the security implementation within ERP software environments. A further increase in time and cost can be considered when attempting to document any of the security subsystem configuration and settings. As is customary in large-scale software implementation projects, the need for both system and end-user documentation is critical. During the development and integration phase of the new system, some form of change management is also required.

Most of the ERP software packages selected for investigation during the course of this study support change management and documentation for the mapping of the business processes to distinct process flows within the ERP system. Hence, the documentation of business process mapping to the configuration of the system is supported in most cases. In a similar fashion, a change management module is available to track changes made to key objects and elements within the system. Unfortunately, none of the systems selected for this study provide any form of change management or documentation support for the security subsystem. It is not known why no change

management system has been implemented for the security subsystems in the ERP systems investigated during the course of this study. The lack of such a feature is considered to be a grave omission, however[6].

The lack of change management may be considered very serious. Without any form of change management documentation, workflow or control, all changes made within the system to effect changes to user master records and access permissions cannot be traced. Though the systems under investigation record the name of the user who effected the last change, there is no way of tracking the change and ensuring it complies with security requirements set out within the organization. Once again, the central nature of security object administration is a cause for concern. Security administrators are often under pressure to provide access to certain functions at short notice. Paper-based change control systems are difficult to maintain and tracking of changes becomes almost impossible.

4 DECENTRALIZING THE APPROACH TO SECURITY

Within traditional ERP environments, the centralized approach to implementing access control and access restrictions enables one or more security administrators to create and maintain profiles, roles and user master records. As has been mentioned above, this approach suffers from a number of problems, most notably that the security administrator cannot and usually does not understand the complexities of the actual business processes within the organization and how these have been mapped to the functionality of the selected ERP software package. To combat this problem and to promote more rigid and adequate security within an ERP environment, it is necessary to deal with complexity within the system as a whole. Figure 2 below indicates the changes from the centralized approach. It is important to notice that the security administrator still performs a management and auditing role, but is no longer concerned with the detail of the individual user requirements.

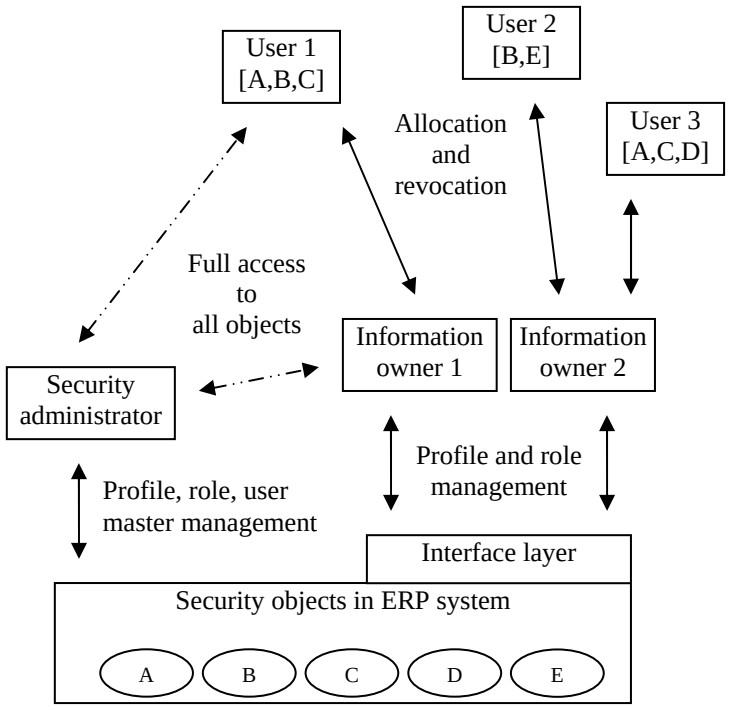

Figure 2. Decentralized security within an ERP environment

The decentralized approach suggested in Figure 2 ensures that the technical complexity associated with the creation and allocation of security objects is removed. This is achieved by presenting the identified information owners with an interface layer. The interface layer hides technical complexity from the information owners and presents them with only the necessary data relevant to their area of the system and business. As the control of the overall security subsystem must rest with one or more individuals, the security administrator has the usual, full access to all security objects at the most detailed, technical level. This permits the security administrator to tailor and allocate objects for the specific use of the information owners. Once the information owners are satisfied that all required security objects are accessible to them, the require security objects can be tailored and allocated to users within the business sphere of the relevant information owner. In a more advanced model, the security administrator may not be able to allocate any security objects to any users, but rather prepare security objects for use by the information owners. This addition to the decentralized model would ensure a higher degree of separation and ensure that no access to the system could be gained through the security administrator.

4.1 Dealing with complexity

ERP software packages provide integrated functionality across an entire enterprise. Generally, such software packages may be deployed in various industries, across various disciplines and across various countries. This makes ERP systems rather complex in their architecture and feature set.

Due to the complexity of modern ERP software packages, the ability to fully understand all functionality offered by the software package is virtually impossible. Specialist areas of expertise are thus created. The creation and maintenance of access restrictions and their related objects within the system may be considered such a specialist area of expertise. The ability to create and maintain a security infrastructure for areas of the business that are merely described and documented by knowledgeable members of the organization should not be considered adequate. To state this problem in a different way: the security administrator has the technical knowledge to create and maintain security related objects within the system, but is unable to fully grasp the complexities of the business processes and all related nuances. Conversely, the business owner has the necessary knowledge relating to the business processes, risks and areas of concern, but does not possess the technical knowledge to create and maintain the necessary security settings and objects within the system.

The problem of complexity is further exacerbated by the fact that the communication from the business owner to the technical security administrator may be unclear or misunderstood by either party. This results in a business owner and a technical security administrator who both believe that adequate security and access restrictions have been placed within the system. Any miscommunication or mistakes may be detected only once fraud has occurred or an audit on the system has taken place. It should be clear that such a situation is not acceptable.

A proposed method of dealing with this complexity is to permit each business owner to become responsible for the setting up of access restrictions within the system. However, the business owner should be able to perform such activities only within the area that the business owner is responsible for. In providing this feature, the level of security could be increased dramatically and the timeframe for its implementation reduced significantly. The reason for this is that the business owner is aware of all required access restrictions and security requirements within his sphere of responsibility.

The above should indicate that complexity can be dealt with by permitting the business owner the means to ensure adequate access restrictions to all objects and data within his sphere of responsibility. However, the topic of responsibility should be touched upon briefly.

4.2 Improving responsibility and accountability

The complexity of creating various roles, profiles and user master records within an ERP system can be eased by permitting each business owner to take over the function of allocating and creating security objects within the ERP system. An important side-effect of this concept is that of increased responsibility and accountability.

To elaborate on this concept, consider the traditional approach to implementing access restrictions within an ERP environment: a business owner determines who requires a certain type of access to complete certain tasks. Often, such requirements are documented by the business owner in a spreadsheet, indicating required functions per user. This set of documentation is passed on to the security administrator, who determines the requirements and configures appropriate security objects within the system. Such security objects are often roles or profiles. Once these are created, the security administrator allocates them to the appropriate users' master records within the ERP system. Some unit testing may be performed.

Though the above procedure seems to be acceptable, it should be considered that the business owner is generally not aware of the technical details of the security objects allocated to the users. This may have serious implications if the business owner is to be held accountable for an action taken by one of the users who have been allocated one or more of these security objects. If one of the users is able to perform a task or function that results in fraud, for example, the business owner may be held responsible for allocating an inappropriate function to that user. However, the security administrator may have made a mistake or added the extra function to cater for functionality required elsewhere. Clearly, the business owner cannot be held accountable for a mistake made by the security administrator. It is vitally important therefore, to ensure that the correct access authorizations are allocated to the correct users and entities within the system. In an ideal model, the accountability for the allocation of access rights to the system and its objects should rest with only one person. This person should be the identified information owner for a section of the business.

By providing the ability of creating and maintaining security objects to the business owner, accountability and responsibility is more visible within the organization. The occurrence of fraud due to the inappropriate allocation of a function or transaction to a certain user can now be traced directly to a single source.

4.3 Moving towards information ownership

The concepts discussed briefly above contribute most towards creating a decentralized security infrastructure within an ERP environment. Although numerous other concepts can be mentioned, these are beyond the scope of this study. Reduction of complexity and increased accountability and responsibility are considered the most important elements for the move towards increased information security with ERP systems. The concept of permitting individuals within the business to manage and maintain their own information security is termed information ownership. This concept will be expanded upon in the remainder of the paper.

4.4 Decentralizing without the need for more technical knowledge

It is necessary to stress the importance of reducing the need for more technical knowledge. The paragraphs above have dealt with the requirement to reduce complexity and to increase accountability. This cannot be accomplished if the security administrator's task is simply handed over to the individual business owners.

The business owners should not have to be taught the technical implementation details of creating roles, profiles and user master records. Instead, the technical complexity should remain

with the security administrators, who maintain a support function. The creation of roles and profiles, together with the allocation of these to user master records should take place in a user-friendly environment. Ideally, the business owner should simply allocate functions and access requirements to a user by means of “point and click”. If required, a more complex access requirement could be created by the security administrator for allocation by the business owner.

The need to make the security subsystem user-friendly and less technical for the business owners cannot be overstated. In fact, this can be considered the most important aspect for supporting the concept of information ownership.

5 INFORMATION OWNERSHIP

Though the term information ownership has been in use for some time, the concept of information ownership has not moved into the sphere of ERP software systems. The advantages of ERP software packages stemmed primarily from their ability to integrate all data within the organization, to deliver real-time results and reporting and to make specific functionality available to the user at the desktop level. Unlike various business functionality and automation of business processes, the implementation of security is still considered a centralized function.

In stark contrast to the adaptability and flexibility of being able to configure the ERP software package to the needs of the business, the configuration of security related objects is completed by technical staff. Mention has already been made of the complexity of ERP software packages[7]. This complexity is necessary for the software package to be adaptable to different industries and legal requirements. The ERP software packages investigated during the course of this study provide full support for the configuration of the software to adapt it to support various business processes. Similar functionality for the configuration of security objects is missing. In fact, all security-related objects are generally grouped together and are not easily distinguished from one another. The ability to document the necessary access restrictions and security objects is hampered by very technical naming conventions. In ERP systems targeted at organizations with a smaller user population, the configuration of the security subsystem is often fairly trivial, offering the security administrator very little flexibility. It is clear that concept of information ownership is not supported by the ERP software packages investigated during the course of this study.

5.1 Supporting information ownership

The support for information ownership within the ERP software packages investigated during the course of this study is not natively possible. In other words, extensive changes to the user interface and possibly data structures would be necessary. The primary reason for this is the architecture employed by the software vendors to support their centralized security subsystems.

All the ERP systems under investigation employ a repository within which all information is stored. This pertains to data specific to the software system, customer information and security information. To enable the user master records to be filled with the necessary access and authorizations, various security objects need to be configured by a security administrator. Though the method and implementation differs from one vendor to another, the procedure can be unified and described as follows: each object within the ERP system has a certain access entry associated with it. To gain access to such an object, the user master record must contain an entry that corresponds to the access entry. Within the code of the ERP software, certain routines determine whether or not the user is permitted to access certain information or whether a function may be executed. If the entry within the user master record matches with the required entry, the user is permitted to continue.

The user master record is stored in one or more tables. All access entries required by the objects within the system are also stored in tables. Due to the fact that an ERP system spans many different functional areas, also known as modules, numerous access entries exist. General ledger, sales, materials management and asset management are examples of common modules within an ERP system. The centralized nature of the security subsystem permits the security administrator to allocate as many different access entries as required. This is regardless of the functional area under consideration. The inability of an ERP system to natively support information ownership can be summed up as follows: there is no support for determining what access entries the current user is permitted to allocate. In the centralized model favoured by the ERP software vendors, the information owner would be able to allocate all access rights to all modules within the system. The reason for this is that the system is unable to determine which area of the system the information owner is responsible for. Clearly, a mechanism is required to enforce such validation.

5.2 Validating information owners

The validation of information owners is important to support information ownership. For information ownership to be viable, the system must allow only certain users to be able to allocate certain tasks and functions. If this is not the case, the architecture reverts back to a centralized security configuration where a single user is able to allocate all access authorizations across all modules of the system.

To validate information owners, a simple check has to be performed by the security subsystem before any access entries for objects may be allocated. Thus, the system has to ensure that the information owner is in fact an information owner and that the information owner is indeed authorized to allocate the security objects in question. As multiple information owners exist within the system, these two checks would have to be performed by the system. In effect, this becomes a double authorization check within the code of the ERP system, but is in effect only when the information owner is busy with the creation of roles and profiles. To ensure correct operation, the system should be capable of displaying only those objects pertaining to the module or area within which the information owner is supposed to be working.

Equally important is the scope of users to which the information owner may allocate roles and profiles. In a similar fashion to the way in which the system should restrict access to only the access entries available for allocation to roles and profiles, so too should access be restricted to only the users reporting to the information owner. The information owner may not allocate access authorization to users outside the scope of his sphere of responsibility.

5.3 Shared responsibility

Once the information owner is able to allocate only certain access rights to certain users, the concept of information ownership is assured. However, a potential pitfall is the case where various users may require access to functions or information that exist within the sphere of another information owner. As an example, a user creating a sales order may require access to determine the credit worthiness of the customer. In this example, two information owners may have to be involved: the information owner for the sales module and the information owner for the financial module. For this to be possible, a mechanism must be in place to permit the information owner for the sales module to allocate a certain function or task to a user in the financial module. Similarly, a user within the financial module may require access to a function or task within the sales module.

The mechanism to achieve this is required due to the integration of functionality and data within an ERP system. A number of tasks may require some form of cross-module integration within the ERP system. The simplest solution to the problem of shared responsibilities is to permit each information owner to expose basic functions to other information owners. Within the ERP

system's security subsystem, such a function could be possible by permitting an information owner to create basic access rights for common functions that may be required by other users in the system. If a function needs to be executed by a user under the responsibility of another information owner, for which the access rights have not been created, a formal request would have to be made. It would be up to the information owner to determine whether or not such access would be granted. In this way, the responsibility and accountability of the information owner would be maintained and assured.

6 ADVANTAGES OF INFORMATION OWNERSHIP

From the discussion above, the advantages of the approach to security within ERP software systems should be clear. This is especially so when compared to the traditional centralized approach favoured by current ERP software vendors.

In summary, the primary advantages of the information ownership approach to enhancing security within ERP software systems are:

- a reduction of complexity;
- the ability to increase responsibility and accountability within the organization;
- a faster implementation time by providing decentralized access to security objects;
- to improve the quality of the security configuration as a whole;

7 ERP SYSTEM SUPPORT FOR INFORMATION OWNERSHIP

It has been stated in a previous paragraph that existing ERP software packages cannot simply be retrofitted to support the information ownership concept. The primary reason is the requirement for a different architecture to support the information owner and the mechanism of allocating access rights to users assigned to the information owner's group.

In addition to the requirement for an extended architecture for the support of the decentralization of the security objects, an integrated change management and documentation module is required. Ideally, the change management and documentation module should be integrated with a workflow engine that permits automatic routing of tasks and requests from one information owner to another. A detailed discussion of these aspects is beyond the scope of this paper. A complete and ideal model for a security subsystem that supports all aspects of information ownership to enhance security within ERP software packages is under development[6].

7.1 Corporate governance in support of information ownership

A strong case for the information ownership concept is the current implementation of various legal requirements in different countries of the world. In the United States of America, the Sarbanes-Oxley act has become increasingly important. In Europe and the United Kingdom, the Basel 2 Accord and Turnbull Report respectively are gaining increasing importance[8]. Though many of the world's largest and most influential organizations make use of ERP technology, very basic support for the segregation of duties issues required by legislation is present in existing ERP systems. Segregation of duties is seen to be one of the most important aspects to prevent fraud and heighten security[9].

8 CONCLUSION

This paper has briefly described some of the problems encountered when implementing security within existing ERP software packages. An alternative has been proposed, that permits the formerly centralized security architecture to be decentralized. The concept of information ownership was briefly explained, and its importance highlighted. By supporting information ownership, the security of ERP software packages can be increased and improved.

9 REFERENCES

1. Joseph R. Dervaes, Internal Fraud and Controls, Washington Finance Officer's Association, 48th Annual Conference, 19 September 2004
2. SAP AG, SAP R/3 Online Help CD-ROM, 2003
3. Oracle Corporation, Oracle Applications System Administrator's Guide, Release 11i, Volume 1, 2002
4. Microsoft Business Solutions, Installation and System Management Manual: Navision Attain, Navision, 2002
5. Microsoft Corporation, Axapta System Administration Guide, Microsoft Business Solutions, 2002
6. M. Hertenberger, Prof. S.H. von Solms, Ph.D. study in progress: "Security in ERP environments", Rand Afrikaans University, Johannesburg, South Africa, 2004
7. K. Vuppula. BW security approaches, http://www.intelligenterp.com/feature/2002/12/0212feat1_1.shtml, 2002
8. P. Manchester, Financial Times, 12 November 2003
9. Elizabeth M. Ready, Emerging Fraud Trends, State of Vermont, 2003
10. H. Silverstone and M. Sheetz, Forensic Accounting and Fraud Investigation for Non-Experts, John Wiley & Sons, December 2003