Modeling Access Rights Using the CRUD Security Cube: An Extension Incorporating Time

Michael R. Collins, Ph.D., mcollins@highpoint.edu
Department of Management
Phillips School of Business
High Point University
High Point, NC 27262

*Dale L. Lunsford, Ph.D., **Dale.Lunsford@usm.edu***
Department of Business Economics and Decision Sciences
The University of Southern Mississippi
730 East Beach Blvd.
Long Beach, MS 39560

Modeling Access Rights Using the CRUD Security Cube: An Extension Incorporating Time

ABSTRACT

This paper builds on the CRUD Security Cube (Lunsford & Collins, 2008) research by incorporating time as an important variable in the process. In this paper we identify why accounting for time is important and how time can be incorporated and accounted for within the CRUD security cube approach to securing and providing appropriate permissions to objects.

INTRODUCTION

Defining access rights is a challenge in many settings. Since a database often serves as the foundation for information systems, proper specifications at the database level can ensure proper access rights exist within the system. How do organizations set and maintain user and group access rights to information systems in general and within databases specifically? Turnover, promotions, job and task shifts are just a few of the situations that arise in maintaining an up-to-date set of security and access rights for users and groups within organizations today. This paper describes a database implementation of access rights using the CRUD Security Cube (Lunsford & Collins, 2008) and incorporates the time dimension into the CRUD proposed security cube model.

Access Rights

Although the nature of an access right varies from system to system, most contemporary systems provide some mechanism for managing access to resources. Access rights, also known as permissions or privileges, define the types of access a user or group has to a securable object. In many systems, access rights apply to either users or groups. In Unix systems, access rights apply to an object's owner, a group, and the world (December, 2008). In Windows systems using the NT File System (NTFS), access rights apply to users and groups (Melber, 2006). The target resources for access rights include directories and files, devices, executables, as well as other objects (Changing Access Security on Securable Objects, 2008). Common access types include full control, modify, read & execute, read, and write under NTFS (Melber, 2006; Eckel, 2007) and read, write, and execute under Unix (December, 2008). NTFS offers advanced mechanisms for access rights, including inheritance and the ability to deny access (Melber, 2006; Mullins, 2006; Eckel, 2007). Additionally, under NTFS the specification of access rights is either explicit or inherited. Finally, NTFS provides the ability to deny a user or group any particular access type.

THE CRUD SECURITY CUBE

The traditional CRUD matrix provides a method for identifying the types of access system processes have to data objects. The CRUD Security Cube adds a user/group dimension to the CRUD matrix (Lunsford & Collins, 2008). This dimension documents

the access rights for users or groups to processes and data. Analysts may use the CRUD Security Cube to specify security for information systems, including any setting where the user employs specific programs to access data objects.

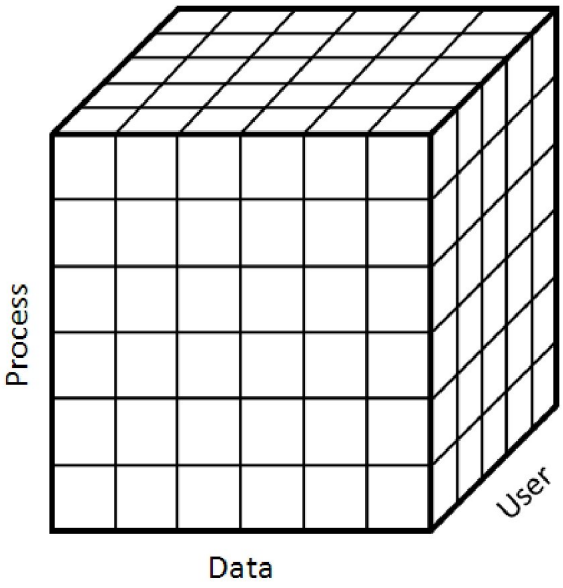
In this paper, we propose an extension of the existing proposed security cube to include an incorporation of time as a valid and important dimension through which organizations would want to control users or group's access and privileges to processes and data objects.

A Time Dimension Example

The CRUD matrix assists database administrators in mapping out usage access for databases within an organization. Working from the CRUD Security Cube extension, this paper proposes the incorporation of time within the security cube. Using time as a fourth dimension, while hard to draw, is very important conceptually. Most organizations have constraints and policies in place that require strict attention to what processes and objects are available to what users and groups and to what extent those privileges are granted. The question we ask in this paper is do those privileges remain the same for all points in time? Stated another way, would a particular user or group have access to (create, read, update, or delete) a process or data object at one point in time and not have access to that same object or process at a different time? With many organizations controlling when access is granted is as important as the granting of the access itself. Many situations call for the granting, ungranting, and granting again of access to a process or object.

Using time as another dimension to the proposed CRUD security cube this need to restrict and allow access across time can be accomplished. Presented in Figure 1 is the original security cube as proposed in (Lunsford & Collins, 2008).

FIGURE 1: CRUD SECURITY CUBE



Incorporating the Time Dimension

Adding time to the CRUD security cube would in effect potentially add a very large number of cubes to represent the point in time the access is granted or removed from a user or a process. This unit of time could be months, weeks, days, hours, minutes, or even seconds depending on the needs of the organization. Imagine if you will a “long row” of cube after cube after cube with each cube representing the setting for the CRUD security cube at a particular point in time.

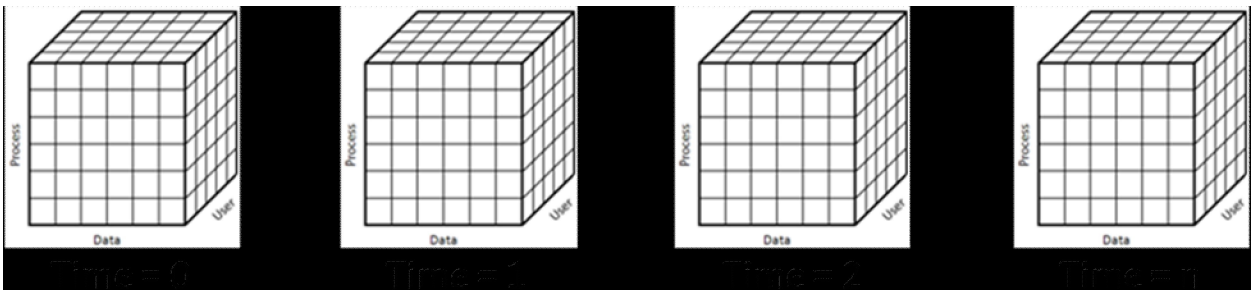
This incorporation of the time dimension could be implemented in a database by adding a time parameter and having the security management program scan the security table for restrictions or grant requests based on points in time. This security management program would scan the security table several times a second for changes to the security settings.

FIGURE 2: GROUPS, PROCESSES, AND DATA OBJECTS

Groups	Processes	Data
Group One	Maintain Inventory	Customer Information
Group Two	Invoice Customer	Vendor Information
Group Three	Pay Vendor	Product Information

Figure 3 depicts the CRUD Security Cube with the time dimension.

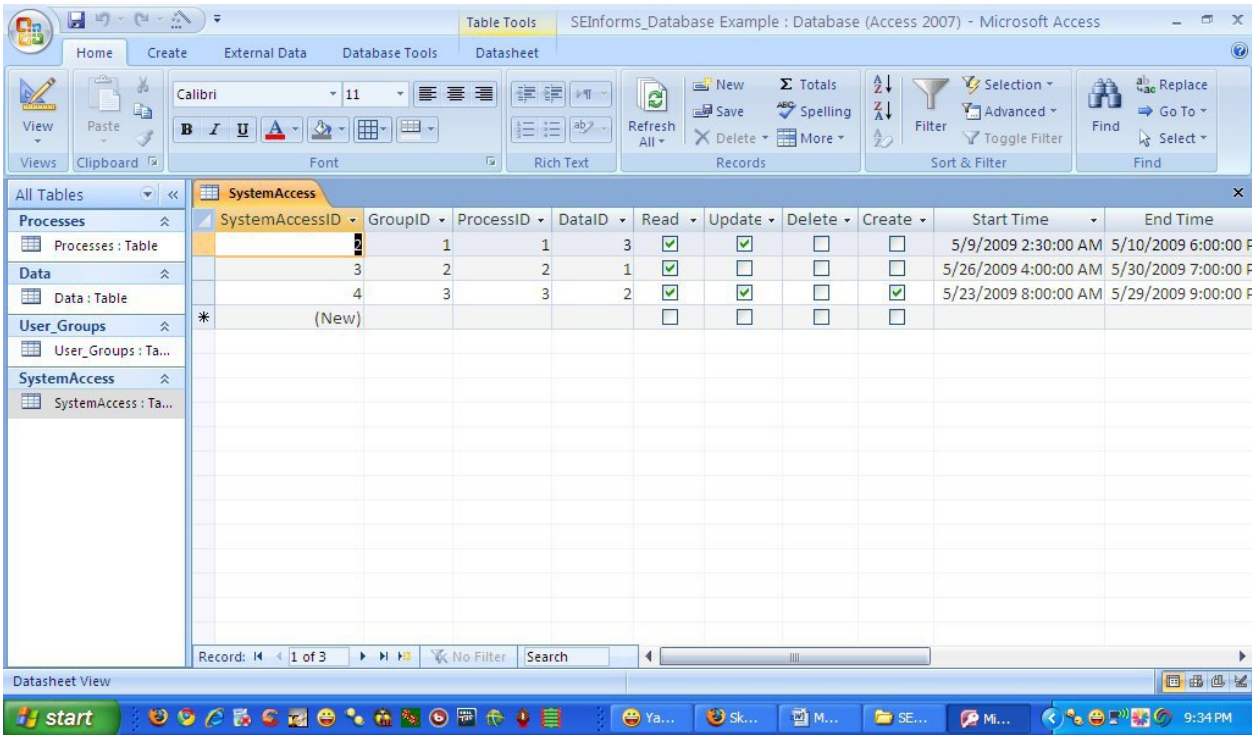
FIGURE 3: CRUD SECURITY CUBE WITH TIME DIMENSION



As you can see from the cube representation in Figure 3, the cube allows a database

administrator to break down individual access rights by group, within a process, for specific data over time. This information can then be entered into a database and updated as needed. Once the database is updated with the information a program can be written to pull the data and settings from the database and update the security and access rights for groups and users automatically. A snapshot of the system access table would look similar to Figure 4.

FIGURE 4: MICROSOFT ACCESS IMPLEMENTATION



Using this system access table presented in Figure 4, the groups or users documented access and security privileges could be extracted and updated in a separate database using Oracle, SQLSever, MySQL, or just about any other SQL-based DBMS on the market today.

In addition to enabling the specification of a time-based access constraint, the addition of the time dimension also enables the security manager or auditor to view a historical record of the access privileges at any point in time. This could prove valuable when investigating suspected inappropriate access to information or programs.

Extensions to this research

Extensions to this research could include additional proof-of-case scenarios that show the versatility of this approach to apply to any type of information system access rights' settings. In this paper we have shown a proof-of-concept example of how the CRUD security cube, incorporating a time component, could be implemented within a database management systems environment. The approach proposed in this paper could be used to automate the

setting of security and accessibility settings for objects with respect to data within individual processes and with respect to groups or individuals of an organization over any period of time.

REFERENCES

[1] *Changing Access Security on Securable Objects.* (2008, February 14). Retrieved February 26, 2008, from MSDN: http://msdn2.microsoft.com/en-us/library/aa384905(VS.85).aspx

[2] December, J. (2008, January 21). *Permissions.* Retrieved February 22, 2008, from December.com: http://www.december.com/unix/tutor/permissions.html

[3] Eckel, E. (2007, January 22). *How do I... Secure Windows XP NTFS files and shares?* Retrieved February 7, 2008, from TechRepublic.com: http://articles.techrepublic.com.com/5100-10877_11-6152061.html

[4] Lunsford, D. L., & Collins, M. R. (2008). The CRUD Security Matrix: A Technique for Documenting Access Rights. *7th Annual Security Conference.* Las Vegas, NV.

[5] Melber, D. (2006, May 3). *Understanding Windows NTFS Permissions.* Retrieved January 25, 2008, from WindowsSecurity.com: http://www.windowsecurity.com/articles/Understanding-Windows-NTFS-Permissions.html

[6] Mullins, M. (2006, June 15). *Windows 101: Know the basics about NTFS permissions.* Retrieved June 19, 2006, from TechRepublic.com: http://techrepublic.com.com/5102-1009-6084446.html