Kommunalsoftware auf dem Prüfstand[1]

(Auszug aus der Veröffentlichung im OKKSA-Newsletter Nr. 18 vom 04.12.08)

Teil 8: Keiner für alles! Filigrane Zugriffsrechte als Praxisbremse? (von Dr. Uwe Schwochert)

Nachdem im Teil 7 der Serie vor allem Programmmechanismen zum Zugriffsschutz und zur Passwortvergabe betrachtet wurden, geht es dieses Mal um den inhaltlichen Aspekt. Also darum, was denn eigentlich vor wem geschützt werden soll und wie mit Hilfe des Programms eine funktionierende Benutzerverwaltung aufgebaut werden kann.
Im Mittelpunkt der Betrachtung stehen dabei wie immer finanzwirksame Programme.

Der Grundanspruch ist relativ klar:

[FÜ4.15] *Das Programm gestattet die übersichtliche Vergabe individuell differenzierter Zugriffsrechte der Benutzer für die einzelnen Programmfunktionen und -aufgabenbereiche.* [MUSS-Kriterium]

Eine der ersten Fragen in diesem Zusammenhang ist die nach den Zugriffen des Administrators selbst: Darf er alles?
Die Diskussion im Fachgremium zu diesem Punkt ergab, dass das Programm unbedingt auch die Einrichtung eines reinen Administrator-Zugangs ermöglichen muss, der keinen Zugriff auf die fachlichen Programmfunktionen hat:

[FÜ4.15a] *Für Aufgaben der Zugriffsverwaltung auf Programme, Programmfunktionen, Daten und Datenbereiche und zur entsprechenden Verwaltung der Programmbenutzer gibt es spezielle Administratorzugänge zum Programm. Diese unterliegen ihrerseits dem Zugriffschutz.* [MUSS-Kriterium]

Nun könnte man einwenden: Wie sinnvoll ist eine Beschränkung des Administrators, wenn er doch jederzeit für sich selbst auch einen fachbezogenen Zugang einrichten könnte? Das Programm kann ja nicht unterscheiden, wer da gerade in welcher Rolle vor dem Bildschirm sitzt.
Hier kommt ein wichtiger Aspekt der Benutzerverwaltung zum Tragen: Es geht nicht nur darum, einzelnen Nutzern Rechte zuzuweisen oder vorzuenthalten, sondern auch die Vergabe der Rechte zu dokumentieren und zu protokollieren:

[FÜ4.22] *Die Änderung von Zugriffsrechten wird durch das Programm protokolliert. Das entsprechende Protokoll ist vor unbefugter Veränderung geschützt.* [MUSS-Kriterium]

Es kann also tatsächlich nicht verhindert werden, dass der Administrator über einen eigens eingerichteten Zugang Zugriff auch die fachliche Datenverarbeitung erhält. Allerdings muss programmseitig sicher gestellt sein, dass das nicht unbemerkt geschehen kann, zumindest nicht ohne explizite Zuweisung einer fachbezogenen Benutzerrolle.

Wichtigste Anforderung an die Benutzerverwaltung im Programm ist jedoch ihre Übersichtlichkeit. Das Problem besteht ja darin, dass ein wirksamer Zugriffschutz im Programm eigentlich nur dann gegeben ist, wenn er sowohl den Datenbankzugriff als auch den Zugriff eines Benutzers auf die Bearbeitungsmasken des Programms reglementiert. Damit kristallisieren sich zwei Arten von Zugriffsobjekten heraus:

[1] Hinweis: Die in der Artikelserie genannten Kriterien stammen aus der aktuellen Fassung des "OKKSA Anforderungskatalogs für Fachprogramme in der Öffentlichen Verwaltung - Teilbereich Fachübergreifende Programmanforderungen", Version 3.1

1. Datenbankobjekte (Tabellen, Abfragen, Berichte, ...)
Hier geht es häufig neben dem Zugriff durch das Fachprogramm auch um den externen Zugriff, z. B. für den Datenexport oder Berichtsgeneratoren. Mit dem Trend, die Zugriffsberechtigung auf dem System-Benutzer-Login aufzubauen ergibt sich die Anforderung, dass bereits das Datenbanksystem hinreichend intelligent Zugriffsrechte verwaltet, um einen differenzierten Zugriff auf die verschiedenen Datenelemente auch außerhalb der Fachanwendung zu gewährleisten.

2. Bedienobjekte des Programms (Bearbeitungsmasken, Menüaufrufe, Module)
Innerhalb des Fachprogramms sind fachliche (und damit an Benutzerrollen bindbare) Abläufe die Kernelemente für die Benutzerzugriffe. Entsprechend soll ein fachbezogener Programmbenutzer auch nur die für ihn relevanten Programmfunktionen benutzen können. Oder, wie häufig umgesetzt, die anderen Menüpunkte und Funktionen gar nicht erst sehen.

Beide, Datenbankobjekte und Bedienobjekte des Programms, haben eins gemeinsam: bei hinreichend umfangreiche Fachverfahren entstehen hier sehr komplexe Strukturen. So wird die Bearbeitung eines Kassenmitarbeiters nicht auf 2-3 Menüpunkte und 4-5 Tabellen in der Datenbank begrenzbar sein. Die Zahlen gehen eher in die Hunderte. Neben der Vielzahl spielen auch die Hierarchien der Zugriffsobjekte eine Rolle: in einem typischen (größeren) Finanzverfahren wird der Kassenmitarbeiter vom Programm her vielleicht mit 5 Programmmodulen zu tun haben. Die haben wiederum vielleicht durchschnittlich 20 vorgangsbezogene Menüpunkte/Bearbeitungsmasken. In diesen gibt es wiederum jeweils 2-3 Optionen, die abhängig von den Benutzerrechten sind.

Genauso bei den Datenbankobjekten: Durch deren Verknüpfung gibt es auch hier keine lineare Struktur gleichberechtigter Zugriffsobjekte, sondern eine komplexes Verzahnung, die das Gesamtdatenmodell widerspiegelt.

Die Sicht des Programmanwenders in der Verwaltung dürfte dagegen eine ganz andere sein: Vor dem Hintergrund von Ämterzuordnungen und Stellenbeschreibungen gibt es klare Regelungen, wer ws darf. Dazu zählen auch die Vorgaben des Gesetzgebers bezüglich des Vier-Augen-Prinzips und der Trennung zwischen Anordnung und Vollzug.

Die Zuordnung der verwaltungsinternen Aufgabenteilung zu den Zugriffsobjekten der Fachanwendung wird vor diesem Hintergrund zu einer entscheidenden Aufgabe bei der Konzeption der Benutzerverwaltung eines modernen Finanzverfahrens.

Im Rahmen der Prüfkriterien des OKKSA-Kriterienkataloges wird dies an mehreren Einzelpunkten festgemacht. Die wichtigsten sind:

[FÜ4.20] *Wenn ein Programm zur gleichzeitigen Nutzung für verschiedene Aufgabenbereiche der Anwender (z.B. Ämter, Bereiche) vorgesehen ist, so können im Programm Benutzertypen (Rollen) angelegt werden, die über bestimmte aufgabenbezogene Zugriffsrechte verfügen. Diesen Benutzertypen können im Rahmen der Rechteverwaltung konkrete identifizierbare Benutzer zugeordnet werden.* [MUSS-Kriterium]

Der Benutzertyp (oder –rolle) ist deutlich mehr, als eine Arbeitsvereinfachung für die Zuweisung von Rechten. Über den Benutzertyp werden Einzel-Zugriffsrechte so gebündelt, dass die aufgabenbezogene Arbeitsteilung in der Anwenderorganisation überhaupt erst abgebildet werden kann. Bei komplexen Verfahren kann dem Administrator vor Ort auch nicht mehr zugemutet werden, dass er selbst diese Bündelung der Einzelzugriffsrechte vornimmt, weil er kaum in der Lage sein wird, die Komplexität und die wechselseitigen Beziehungen der Datenbank- und Bedien-Zugriffsobjekte zu überschauen.

[FÜ4.21] *Im Programm sind standardisierte Benutzertypen (Rollen) entsprechend den Aufgabenstellungen des jeweiligen Anwenderkreises voreingerichtet und dokumentiert.* [KANN-Kriterium]

Das setzt also voraus, dass der Programmentwickler sich vorab mit den typischen Aufgabenfeldern des Anwenders auseinandersetzt und in der Lage ist, passgerechte Rechtebündel zu konfigurieren.

Doch damit nicht genug. Nachdem klar ist, dass nicht das einzelne Zugriffsobjekt, sondern der Benutzertyp bzw. das damit verbundene Rechtebündel entscheidend für die Zuordnung der Benutzerrechte vor Ort ist, wird dieses Rechtebündel zum Betrachtungsgegenstand der lokalen EDV-Administration. Denn es muss mit real existierenden Personen in Verbindung gebracht werden, die wiederum durch Ämterzuordnungen und Stellenbeschreibungen den Aufgaben der Organisation zugeordnet sind. Es ergibt sich also:

Zugriffsobjekte	⇔	Rechtebündel/Rollen	⇔	Aufgaben	⇔	Benutzer
(ZO)		(R)		(A)		(B)

Es wird sichtbar, dass, um konkrete Programmbenutzer (B) mit konkreten Zugriffsrechten zu versehen, eigentlich eine Zuordnung von Rechtebündeln (R) zu Aufgaben (A) aus Sicht der Anwenderorganisation erforderlich ist.
Damit diese Zuordnung nachvollziehbar und prüfbar bleibt, bedarf es mehrerer Voraussetzungen:
Zum einen müssen natürlich die Aufgabenverteilungen innerhalb der Organisation klar geregelt sein. Aus Dienstanweisungen, Stellenbeschreibungen und Strukturplänen muss ableitbar sein, wer wofür zuständig ist.
Zum zweiten müssen die Rollen (R) im Programm diese Arbeitsteilung widerspiegeln können. Hier geht es nicht nur um die hinreichend flexible und intelligente Einzelrechtezuordnung, sondern auch um die Möglichkeit, diese Zuordnung zu dokumentieren. Rollen müssen also auch kommentierbar sein.
Drittens schließlich ist eine übersichtliche Druckbarkeit der vergebenen Benutzerberechtigungen gefordert (vgl. auch Kriterium FÜ1.5). Anforderung an das Programm ist also die automatisierte Erstellung einer Übersicht der vergebenen Benutzerberechtigungen in der Form, dass die fachlichen Zugriffsberechtigungen wieder erkennbar sind. Entscheidend dafür ist eine anwenderorientierte Beschreibung der eingerichteten Rollen und deren Einzelberechtigungen. Nur so kann innerhalb der Organisation eine aufgabenbezogene Rechtezuordnung überwacht werden, ohne dass jeder Beteiligte die internen Strukturen der Zugriffsobjekte im Programm kennen muss.

Wie überall bei der fachbezogenen Anwendungsprogrammierung geht es also darum, Programmstrukturen anwender-kompatibel aufzubereiten und durch ihre übersichtliche automatisierte Dokumentation auch dem, der die Programminterna nicht kennt, eine Kontrolle zu ermöglichen.

Nächster Teil der Serie:
Vom Steueramt zum Adressdealer? – Datenschutz in Finanzverfahren