

Berechtigungsmodellierung im Geschäftsprozessmanagement von KMU

Steffen Bartsch · Carsten Bormann

TZI – Universität Bremen
{sbartsch, cabo}@tzi.org

Zusammenfassung

Die Geschäftsprozessmodellierung für kleine und mittlere Unternehmen (KMU) unterscheidet sich deutlich von der für Großunternehmen, insbesondere im Bereich der Berechtigungsmodellierung. In diesem Beitrag werden Probleme in der leichtgewichtigen Umsetzung von Geschäftsprozessen für KMU erörtert und alternative Vorgehensweisen vorgestellt und diskutiert. Ein neuer Ansatz für die Umsetzung von Autorisierung in Geschäftsprozessen ist das Selbstbedienungsparadigma, mit dem auf sichere Weise die Flexibilität und Ausnahmebehandlung verbessert wird.

1 Einführung

Die Paradigmen der Geschäftsprozessmodellierung und des Workflowmanagements breiten sich in Unternehmen zunehmend aus [GeHS95]. Die Abbildung von statischen Geschäftsprozessen auf Basis von Standard-Software wird seit Jahren für große Unternehmen betrieben. Hierbei werden mit großem Aufwand Geschäftsprozesse erfasst, formalisiert und in der Standardsoftware abgebildet. Dieser Aufwand stellt für kleine und mittlere Unternehmen (KMU) häufig eine zu hohe Hürde dar. Außerdem sind die erarbeiteten Modelle für die flexiblen Arbeitsweisen in KMU zu statisch. Mehr noch als große Unternehmen profitieren KMU von der Dynamik in den täglichen Arbeitsabläufen. Fortschritte in der Softwareentwicklung, wie beispielsweise Web-Technologien und *agile Entwicklungsmethoden* [LBBC⁺02], ermöglichen inzwischen auch KMU, kritische Geschäftsprozesse als Spezialanwendung umzusetzen.

In Unternehmensumgebungen von KMU herrscht zwar aufgrund der geringeren Anzahl an Mitarbeitern und der dadurch bedingten höheren Aufgabenüberschneidungen von Haus aus ein höheres Vertrauen den Mitarbeitern gegenüber. Allerdings gibt es auch in KMU ein Bedürfnis, nicht grenzenlos jedem Mitarbeiter alle Möglichkeiten zu bieten. Mächtige Systeme zur Realisierung von Geschäftsprozessen bringen eine Reihe von Gefahren mit sich. Bedienfehler können zu Datenverlusten oder inkonsistenten Daten führen. Bei einem hohem Maß an krimineller Energie können auch bewusste Manipulationen oder Firmengeheimnisverrat zu Schäden führen. In vielen KMU ist die Integrität der Daten in einem solchen System von höherem Wert als die Vertraulichkeit, denn die verfügbaren Informationen sind für Mitarbeiter auch leicht über andere Wege zu erhalten. Daher sind auch in diesen Umgebungen Zugriffskontrollen für die Mitarbeiter nach Aufgabengebieten wünschenswert.

Auf der anderen Seite widersprechen zu restriktive Zugriffsbegrenzungen aber auch den Anforderungen an die Flexibilität im KMU-Umfeld. Restriktive Berechtigungen können zu Effizienzminderungen im Geschäftsprozess führen: Falls ein Benutzer sich erst an Mitarbeiter mit der entsprechenden Berechtigung wenden muss, erhöht das die Durchlaufzeit des Geschäftsprozesses und damit die Kosten. In der Berechtigungsmodellierung ist diese Abwägung von Interessen also ein zentraler Aspekt.

Verwandte Arbeiten

Für die Autorisierung von Mitarbeitern in Systemen, die Geschäftsprozesse abbilden, gibt es in der Literatur eine Reihe von Ansätze [QuEP94, R.J.01]. In [AtHu96] wird das *Workflow Access Model* vorgestellt, das Petri-Netze zur Repräsentation der Prozesse verwendet. Dadurch lassen sich auf Petri-Netz-Basis die Erfüllbarkeit von *Authorization Constraints* testen.

Eine verbreitete Basis für Berechtigungsmodelle ist die *Role Based Access Control* (RBAC [FeKu92]), die sich gut für die Abbildung von Zuständigkeiten in Geschäftsprozessen eignet. Häufig wird es für dieses Umfeld modifiziert und um *Authorization Constraints* ergänzt, wie beispielsweise in [BeFA99]. Darin werden außerdem Konsistenzprüfungen und Planungsalgorithmen für die logikbasierten *Constraints* vorgestellt. [CaCF01] beschreibt verschiedene Klassen von *Constraints* (temporale, instanz- und vergangenheitsbezogene) und ergänzt das RBAC-Modell um eine Organisationslevel-Unterscheidung der Benutzer.

Das *Task-Role-based Access Control*-Modell in [OhPa03] unterscheidet ebenfalls die beiden Dimensionen „Rolle im Geschäftsprozess“ und „Position in der Unternehmenshierarchie.“ Dabei erfolgt allerdings eine Berechtigung per Aktivitätszuordnung, was gerade in der Ausnahmebehandlung unflexibel scheint. Auch [ThSa98, SMLP05] definieren Rechte über Aktivitäten. [ThSa98] bezieht dabei Abhängigkeiten zwischen Aktivitäten mit ein. Abhängigkeiten zwischen Aktivitäten gibt es in [LiWL04] ebenfalls. Sie werden hier anhand der Zugriffe der einzelnen Aktivitäten auf gleiche Dokumente ermittelt. Auf Basis dieser Abhängigkeiten werden dann statische und dynamische Analysen von Benutzerzuordnungen vorgenommen und Planungsalgorithmen abgeleitet.

W-RBAC in [WaBK03] bietet mittels kontrolliertem Übergehen von logikbasierten *Constraints* anhand von benutzerbezogenen Prioritäten deutlich mehr Flexibilität als die oben genannten Modelle. Allerdings lässt sich mit eindimensionalen Prioritäten selbst für kontrollierte, einfache Umgebungen nicht ausreichend zwischen Benutzern differenzieren.

Auch im Umfeld von Web-Anwendungen werden RBAC-Modelle häufig genutzt [ASKP00, JAGS01, PaSA01]. In [MFWA⁺99] wird für das Web- und Workflow-Umfeld ein einfaches RBAC-Modell vorgestellt, das allerdings Rollendomänen einführt. Dadurch können dieselben Rollen im Zusammenhang mit verschiedenen Domänen vergeben werden. Außerdem werden die Datenobjekte der Aktivitäten besonders gehandhabt und explizit in Aktivitäts-Rollen-Relation eingebunden.

Zwar gibt es wie oben dargestellt eine Reihe von Beiträgen in der Literatur zum Thema Berechtigungsmodellierung im Geschäftsprozessmanagement. Allerdings sind diese häufig auf umfangreiche Workflow-Modelle für komplexe Unternehmensstrukturen ausgerichtet. Nur in wenigen Beiträgen wird die notwendige Flexibilität in Geschäftsprozessen in Betracht gezogen. Eine Behandlung von grundsätzlichen Problemen in restriktiven Modellen in Zusammenhang mit flexiblen Geschäftsprozessen fehlt bisher.

In diesem Beitrag wird daher die Abwägung der Aspekte Flexibilität und Ausnahmebehandlung in Geschäftsprozessen genauer unter dem Gesichtspunkt der Berechtigungsmodellierung in KMU betrachtet. Es werden in Abschnitt 2 Beispiele aus der agilen Anwendungsentwicklung dargestellt, in denen gerade die Berechtigungsmodellierung problematisch war. Erfahrungen in einem Projekt mit einem KMU zeigen, dass häufig initial die benötigten Berechtigungen unterschätzt werden. Da es unmöglich scheint, Fehleinschätzungen auszuschließen, werden auf der anderen Seite im Abschnitt 3 Methoden vorgestellt, mit denen im Betrieb besser mit fehlenden Berechtigungen, vor allem im Ausnahmefall, umgegangen werden kann. Am Schluss folgt eine Diskussion der Ansätze und ein Ausblick auf zukünftige Arbeiten in Abschnitt 4.

2 Berechtigungsmodellierung in KMU

Die Berechtigungsmodellierung in KMU ist auf zwei Weisen besonders. Zum einen sind KMU auf ihre Kernkompetenz, Flexibilität, gerade in Ausnahmesituationen angewiesen. Die Berechtigungsmodellierung muss also explizit diese Flexibilität unterstützen. Außerdem sind die Ressourcen begrenzt, wenn für KMU kundenspezifische Anwendungen entwickelt werden. Die Anforderungsaufnahme muss also wie die gesamte Anwendungsentwicklung schlank und effizient gestaltet werden. Deshalb bietet sich das Paradigma der agilen Softwareentwicklung an [CoLC04]. Durch iterative, schlanke Entwicklungsprozesse werden Fehlentwicklungen vermieden, reale Anforderungen im Laufe der Entwicklung klarer und die Einbindung des Kunden gefördert. Das lässt sich auch auf die Autorisierung in Form einer agilen Berechtigungsmodellierung übertragen.

Trotz Interviews, Prototypen und iterativer Benutzbarkeitsevaluierungen können Ineffizienzen bis zum Produktiveinsatz verborgen bleiben. Die dadurch entstehenden Risiken sind vielfältig. Effizienzminderungen zu Beginn der Verwendung des Systems, die durch unnötige Umwege im Ablauf der Geschäftsprozesse auftreten, können aufgefangen werden, erhöhen aber die Kosten. Ein weiteres Risiko stellt fehlende Akzeptanz der Anwendung dar. Dieses Problem kann durch politischen Spannungen innerhalb des Unternehmens verstärkt werden, wenn schon existierende „Insellösungen“ einzelner Abteilungen abgelöst werden sollen. Fehlende Motivation bei der Mitarbeit an der ganzheitlichen Lösung kann die Folge sein. In der Summe können diese Effekte das gesamte Projekt gefährden.

In dem Projekt im KMU-Umfeld, aus dem diese Erfahrungen stammen, wurde ein Teil-Workflow-System umgesetzt. Wie auch bei dem in [AaWe05] als *Case Handling* bezeichneten Ansatz, stehen hier die Datenobjekte im Gegensatz zum Prozess im Vordergrund. Der Prozesszustand leitet sich aus Datenobjektzuständen ab und ist damit leichtgewichtiger und flexibler in Ausnahmesituationen. In diesem Projekt wurden die für die Geschäftsprozesse notwendigen Berechtigungen von den potentiellen Benutzern zum Teil unterschätzt. Exemplarisch beschreiben wir im Folgenden zwei Beispiele, in denen dieses Muster auftrat.

- Im Wareneingangsprozess fehlte die Möglichkeit, für Mitarbeiter der Lagerverwaltung neue Lieferanten einzupflegen. Über den iterativen Entwicklungsprozess hinweg galt hier die Annahme, alle notwendigen Lieferantendaten seien bereits bei Auftragsannahme in einem anderen Zuständigkeitsbereich in die Stammdaten aufgenommen worden. Tatsächlich ließ sich dadurch die Oberfläche für die Rolle schlank gestalten. Außerdem wurde aus Leitungspositionen diese teilzentralisierte Pflege bevorzugt, da dies Doppelangaben eher verhindere. In abschließenden Evaluierungen ergab sich allerdings die offenbar nicht unübliche Situation, dass Lieferungen vor den dazugehörigen Aufträgen

eingingen. Es hätte also bei der Annahme der Lieferung ein entsprechend berechtigter Benutzer hinzugezogen werden müssen.

- Ein weiteres Beispiel ergab sich im Prozess für manuelle Dokumentation von Qualitätskontrollen. Für die Dokumentation wird auch der Standort der Prüfung aufgenommen. Auch hier gab es die Annahme, dass diese Daten zuvor im Rahmen der Auftragsannahme angelegt worden seien. In der Praxis ergaben sich dann Situationen, in denen Prüfungen zu dokumentieren waren, bevor der dazugehörige Auftrag erfasst war. Aufgrund der begrenzten Berechtigung der Qualitätsprüfer verzögerte sich die Prüfung bis entsprechend berechnigte Mitarbeiter den Standort angelegt hatten.

In beiden Fällen wäre es notwendig gewesen, berechnigte Mitarbeiter ausfindig zu machen. Ohne zusätzliche Hilfsmittel erfordert dies erhebliches Wissen über die Rollendefinitionen.

Aus diesen Unzulänglichkeiten lassen sich eine Reihe von Schlüssen ziehen. Häufig werden in Interviews und Prototyp-Evaluierungen durchaus regelmäßige „Ausnahmefälle“ nicht als relevant betrachtet und nicht erwähnt. Ein Grund könnte sein, dass diese Fälle in informalen Geschäftsprozessen keine besondere Behandlung benötigen oder einfach zu umgehen sind. Außerdem besteht in den Gesprächen mit Entwicklern eine Neigung, den „richtigen“ Prozess zu beschreiben. Gleichzeitig sind in diesen Arten von Entwicklungsprozessen die verfügbaren Ressourcen für eine intensive Analyse der bestehenden Geschäftsprozesse nicht ausreichend, um direkt alle Ausnahmefälle zu beachten.

Die Berechtigungsmodellierung hat in der Entwicklung bereits einen hohen Stellenwert eingenommen. Da sich trotzdem die Anforderungen nicht optimal erfassen ließen, sollten zusätzliche Methoden angewandt werden, um Zugriffsberechtigungen zu modellieren. Ein wichtiger Aspekt ist hier eine klarere Abwägung der Berechtigung nicht nur unter der Maxime der möglichst restriktiven Rechte. Wenn eine hohe Flexibilität in Geschäftsprozessen benötigt wird, muss umgekehrt auch jeweils die Frage gestellt werden, welche Ausnahmesituationen möglich sind und warum bestimmte Aktionen einzelnen Rollen nicht gestattet sind. Gerade in der geschilderten Umgebung in KMU ist Flexibilität eine wichtige Anforderung, denn die Flexibilität stellt häufig ein Alleinstellungsmerkmal für KMU dar. Die erhöhten Risiken können durch die gesteigerte Produktivität gerechtfertigt werden.

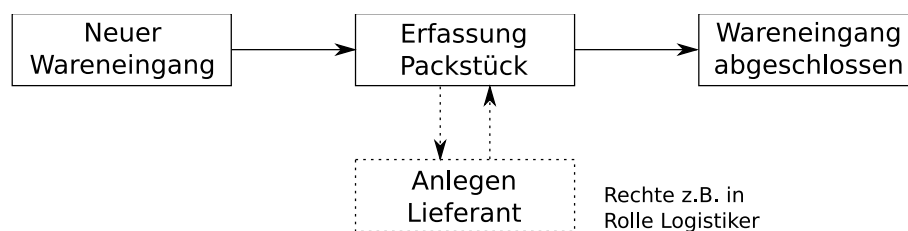


Abb. 1: Beispiel-Prozess mit fehlender Berechtigung im Ausnahmefall

Das oben beschriebene Beispiel der fehlenden Rechte in der Warenannahme ist in Abbildung 1 als Prozessfluss dargestellt. Für die Erfassung von Packstücken der Lieferung ist in Ausnahmesituationen das Anlegen des Lieferanten notwendig. Die Rechte hierfür hat der Lagerverantwortliche jedoch im Normalfall nicht, wodurch ein Kontakt zu einem höher berechtigten Benutzer im Unternehmen notwendig wird. In den folgenden Abschnitten werden Ansätze vorgestellt, um mit dieser Art von Ausnahmesituation umzugehen.

3 Das Selbstbedienungsparadigma

Klassisch werden Berechtigungen in einem hierarchischen Modell in einem *Top-Down*-Ansatz mit dem Paradigma der geringst notwendigen Berechtigung erteilt [HuKI98]. Dies ist zur Begrenzung von Schäden durch Fehlverhalten durchaus sinnvoll, wenn hohe Risiken zu vermeiden und komplexe Strukturen abzubilden sind. Ein anderer Ansatz kann allerdings vorteilhaft sein, wenn Sicherheitsrisiken überschaubar sind. Außerdem sind KMU aufgrund ihrer Struktur unabhängig von Berechtigungen in Systemen auf das Vertrauen in ihre Mitarbeiter angewiesen. In diesen Fällen können die Aspekte der Effizienz, der Bedienbarkeit und der Erlernbarkeit einer Oberfläche im Vordergrund stehen und Vorteile der flexiblen Arbeitsweise des Unternehmens sich auch in der Berechtigungsmodellierung widerspiegeln. Dabei helfen „demokratische“, *Bottom-Up*-Ansätze, wie sie im Web 2.0 auftauchen, die sich analog auch in der Berechtigungsmodellierung anwenden lassen. Mit einer Selbstvergabe von Berechtigungen können Benutzer hier in definierten Grenzen ihre Rechte selbstständig erweitern, wenn dies in Ausnahmesituationen notwendig ist. Dieser Zustand wird im Folgenden als *Ausnahmemodus* bezeichnet.

Die durch den Ausnahmemodus zusätzlichen Risiken ergeben sich aus den erweiterten Rechten für die Mitarbeiter. Diese könnten diese Möglichkeiten für Missbrauch nutzen oder durch ungewollte Fehler Schäden anrichten. Das System bietet eine Reihe von Maßnahmen, um diese mögliche Schäden zu verhindern und, falls sie doch eintreten, zu begrenzen. So werden nur in begrenzten Maße Rechte im Ausnahmemodus zugeteilt. Außerdem sorgt ein detailliertes Protokollieren und Monitoring dafür, dass der Einsatz des Ausnahmemodus bemerkt und nachvollzogen werden kann. Unternehmen können auf Basis dieser Informationen Missbrauch ahnden und Sanktionen einleiten.

In RBAC werden Benutzern Rollen zugewiesen, die der Benutzer aktivieren kann. Im klassischen Modell werden dabei die Rollen so restriktiv wie möglich zugewiesen. Um ein Selbstvergabemodell mit RBAC abzubilden, muss das Modell angepasst werden. In der Rollendefinition wird dann zusätzlich festgehalten, welche möglichen Berechtigungserweiterungen sind, die in Ausnahmesituationen im Ausnahmemodus aktiviert werden können. Zur Begrenzung der Risiken sind Berechtigungen zu destruktiven Aktionen in diesem Modus ausgeschlossen. Das geschieht entweder durch eine Einschränkung der zugewiesenen Rollen, explizite Markierung der Rechte zu diesen Aktionen als „destruktiv“ oder durch Definitionen von Grenzen in *Authorization Constraints* [BeFA99]. Die Risiken können weiter durch die Nachvollziehbarkeit und die Rückgängigmachbarkeit von vorgenommenen Änderungen verringert werden. Details zu diesen Einschränkungen sind im nächsten Abschnitt beschrieben.

Im dargestellten Beispiel aus Abbildung 1 könnte der Rolle des Lagerverwalters als Berechtigungserweiterung die Rolle „Logistiker“ zugewiesen sein. Bei einem entsprechenden Ausnahmefall würde der betroffene Mitarbeiter den Prozess unterbrechen und in den Ausnahmemodus wechseln. Von dort hätte er die Möglichkeit den fehlenden Lieferanten im System anzulegen. Zurück im Normalbetrieb ließe sich der Warenannahme-Prozess wie gewohnt durchführen.

Implementierung

Zur Implementierung des Selbstbedienungsparadigmas werden ein RBAC-basiertes Berechtigungsmodell und Maßnahmen zur Eingrenzung von Risiken vorgestellt. Als erprobtes Grundgerüst für die Berechtigungsmodellierung in der Anwendung dient hier *Role Based Access Control* (RBAC [FeKu92]). Das klassische RBAC-Modell besteht aus den Elementen Benutzer

(*User*), Rolle (*Role*) und Recht (*Privilege*), gezeigt in Abbildung 2. Diese Elemente werden zu Relationen verbunden. Vom Benutzer annehmbare Rollen werden per *can_play*-Relation abgebildet. Grundsätzlich können Benutzern beliebig viele Rollen zugewiesen werden. Den Rollen werden über die *holds*-Relation entsprechende Rechte zugewiesen. Außerdem sind Rollen in einer Hierarchie angeordnet, die über die *is_a*-Relation realisiert wird. So erbt eine höhere Rolle die Rechte der niedrigeren.

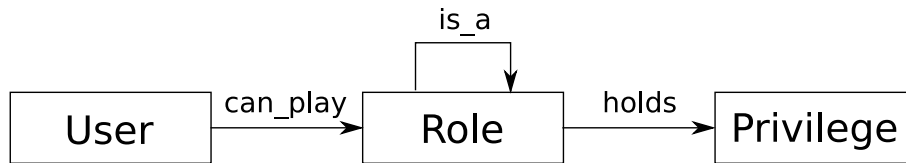


Abb. 2: Allgemeines RBAC-Modell

Für die Verwendung in dem Web-basierten Workflow-System wurde das klassische RBAC-Modell um Aspekte erweitert, die die Ausnahmebehandlung in den Arbeitsabläufen und die Verbindung mit einzelnen Aktivitäten fördert. Entgegen der in der Literatur zu findenden Ansätze, Rechte über Aktivitäten zu definieren, wurden hier Rechte explizit Rollen zugeordnet. Dadurch bleibt ein höheres Maß an Flexibilität erhalten, um auch Zugriffe außerhalb der Prozesse zu ermöglichen. Wie in Abbildung 3 dargestellt, wurde das oben beschriebene klassische Modell um das Element Aktivität (*Task*) erweitert. Mit der Relation *requires* lassen sich nun die für eine konkrete Aktivität notwendigen Rechte der Aktivität zuordnen. Damit ist eine Anbindung an das Workflow-Modell geschaffen, ohne Flexibilität im Berechtigungsmodell zu verlieren.

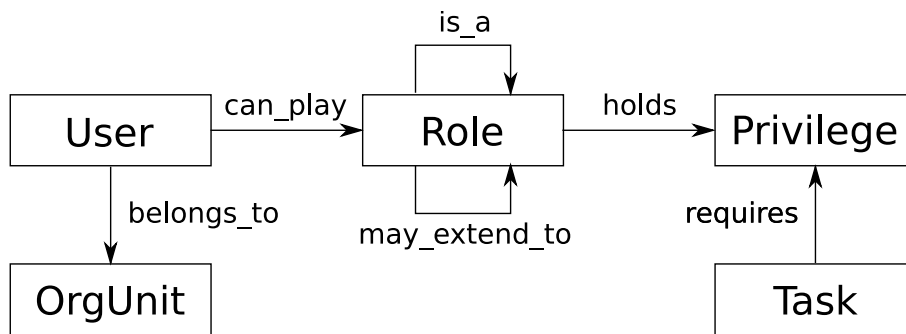


Abb. 3: Erweitertes Rollenmodell

Als Ergänzung für die Behandlung von Abläufen durch dieses Berechtigungsmodell ist außerdem das Element Organisationseinheit (*OrgUnit*), wie auch in [WaBK03], ergänzt worden. Dieses definiert mit der Relation *belongs_to*, welchem Bereich des Unternehmens ein Benutzer zuzuordnen ist. Daraus erwachsen zwar keine zusätzlichen Berechtigungen, es lassen sich aber in *Authorization constraints* Begrenzungen anhand dieser Zuordnung einrichten.

Die dritte Erweiterung zielt auf die Ausnahmebehandlung ab. Mit der Relation *may_extend_to* kann rollenspezifisch festgelegt werden, wie weit die Kompetenzen des aktiven Benutzers im Ausnahmefall ausgeweitet werden dürfen. Aus der aktuell gespielten Rolle kann ein Benutzer damit also zu der über diese Relation verknüpfte Rolle wechseln. Das begrenzt auf einfache, wartbare und effektive Weise das Selbstbedienungsparadigma.

Diese Rollenerweiterung für den Ausnahmemodus kann allerdings leicht auch durch Fehler in der Berechtigungsmodellierung, gerade bei Nachbesserungen in späteren Iterationen, dazu führen, dass doch destruktive Vorgänge zugänglich werden. Drei Ansätze werden verfolgt, um diesem Kontrollverlust entgegenzuwirken. Zum einen wird Rechten die Eigenschaft „destruktiv“ zugewiesen, falls diese jeweils destruktive Auswirkungen haben. Sie werden dann im Ausnahmemodus den Benutzern nicht zur Verfügung gestellt.

Als allgemeiner Schutz vor Datenverlusten werden die Daten in der Datenbank versioniert vorgehalten. Bei ungewollten Änderungen im Ausnahmemodus lassen sich die Änderungen damit direkt wieder rückgängig machen. Allerdings wird vor dem Rückgängigmachen dieses Vorgangs der daraus entstehende Zustand auf Inkonsistenzen untersucht. Falls in der Zwischenzeit auf diese Änderung weitere Aktionen aufbauten, kann der Weg zurück für eine einzelne Änderung unmöglich sein. Hier müssen die Schäden durch eine entsprechende Änderung gegen den Aufwand, diese zurückzunehmen, abgewägt werden. Es kann allerdings garantiert werden, dass die ursprünglichen Daten noch zugreifbar sind. Aber nicht nur für diese Fälle ist die Versionierung vorteilhaft. Auch im Regelbetrieb ist eine Rückgängig-Funktion vorteilhaft und steigert die Robustheit des Systems signifikant.

Als dritter Ansatz wird eine erhöhte Transparenz bei der Rollenzuordnung angestrebt. In einem Simulationsmodus können geplante Änderungen durchgespielt werden und für einzelne Benutzer die dadurch erlangten Rechte angezeigt werden. Dabei wird jeweils zwischen den Rechten im Normalbetrieb und solchen im Ausnahmemodus differenziert.

Für das Monitoring der Ausnahmesituationen findet eine besondere Protokollierung dieser Aktionen statt. Zusätzlich zu der grundsätzlichen Aufnahme der vorgenommenen Änderungen am System im Normalzustand wird also der Übergang in den Ausnahmemodus explizit festgehalten. Jede in diesem Modus erfolgte Aktion ist im Protokoll dann entsprechend markiert. Zusätzlich erscheint beim jeweils Verantwortlichen in der Nächste-Aktivitäten-Darstellung ein Hinweis auf den Vorgang. Gegenwärtig ist dieser Verantwortliche für das System und die organisatorische Einheit bestimmt, der der Benutzer zugeordnet ist.

Falls notwendig, kann im Detail betrachtet werden, welche Aktionen in dem Modus vorgenommen wurden und ob beziehungsweise wann dieser Modus wieder verlassen wurde. Nachfragen zu dem Vorgang können direkt aus dem System heraus an den jeweiligen Benutzer gerichtet werden. Um dem vorzugreifen, kann beim Übergang in den Ausnahmemodus optional eine Begründung angegeben werden. Diese hilft Verantwortlichen, die Situation einzuschätzen und unter Umständen Änderungen am Berechtigungsmodell vorzunehmen.

4 Diskussion und Ausblick

In diesem Beitrag wurden Ansätze vorgestellt, um die Flexibilität in der Ausnahmebehandlung in dem Maße zu bieten, wie sie für KMU notwendig ist. Zentraler Aspekt ist ein Ausnahmemodus, der Benutzern erlaubt, in Ausnahmesituationen auf kontrollierte Weise vorübergehend ihre Rechte zu erweitern. Dieses Selbstbedienungsparadigma hat allerdings auch potentielle Nachteile. Am schwersten wiegt sicherlich das Sicherheitsrisiko durch die erweiterten Rechte jedes Mitarbeiters. In jedem Fall müssen die Risiken für Angriffe von Innen, wie zum Beispiel Betrugsmöglichkeiten, in einer Bedrohungsanalyse erfasst und abgewägt werden. Daraus kann sich ergeben, dass einzelne, kritische Geschäftsprozesse davon ausgeschlossen werden.

Als weiterer Aspekt steigt die Anforderung an die Nutzungsoberfläche der Anwendung, denn

Benutzer müssen sich auch in unbekanntem Rollen mit begrenztem Lernaufwand zurechtfinden. Außerdem muss in der Ausnahmesituation die benötigte Rolle erst identifiziert werden, wozu entweder Erfahrung oder Unterstützung durch die Oberfläche vorhanden sein muss. Eine Akzeptanz des Ausnahmemodus muss unter Umständen erst erzeugt werden. Insgesamt ist also ein erhöhter Lernaufwand notwendig, der durch eine verbesserte Benutzungsoberfläche ausgeglichen werden muss.

Wie im Beispiel des Unix-Werkzeugs *sudo* kann der Wechsel der Rolle allerdings eine besondere Aufmerksamkeit und Sorgfalt in der unbekanntem Rolle erzeugen. Damit sind Fehler unwahrscheinlicher. Außerdem wurden mehrere Methoden vorgestellt, um die Wahrscheinlichkeit von Fehlentwicklungen zu verringern. Destruktive Vorgänge werden grundsätzlich von Rechteerweiterungen ausgenommen und Versionierung ermöglicht ein Rückgängigmachen der ausgeführten Aktionen. Effektives Monitoring verhindert das Übersehen von problematischen Aktionen. Durch die zusätzlichen Möglichkeiten für den einzelnen Mitarbeiter im System wird auf der anderen Seite eine höhere Identifikation und stärkeres Verantwortungsgefühl hervorgerufen. Dadurch kann sich die Akzeptanz des Systems erhöhen. Außerdem ist gerade das ursprüngliche Ziel, die Flexibilität und Effizienz im Alltag der Ausnahmesituationen zu gewährleisten, ein wichtiger Vorteil. Ohne diese Fähigkeiten riskiert ein solches System nur im begrenzten Rahmen eingesetzt zu werden und damit nicht das gesamte Potential auszuschöpfen.

Trotzdem ist das Selbstvergabeverfahren nicht in jeder KMU-Umgebung und jedem Prozess-element einsetzbar. Der Einsatz des Ansatzes ist außerdem vom spezifischen Geschäftsfeld des Unternehmens abhängig. Die konkreten Effizienzsteigerungen für spezifische Einsatzfelder zu messen, ist Teil zukünftiger Arbeit. Dadurch sollte sich dann eine Abstufung und Einordnung von einzelnen Prozessen bezogen auf die Risiken ergeben.

Trotz der erhöhten Flexibilität der Geschäftsprozesse durch das Selbstbedienungsparadigma können weiterhin Hürden in den Abläufen nicht ausgeschlossen werden. Deshalb können die bestehenden informellen Ad-Hoc-Prozesse als Vorbild für eine alternative Ausnahmebehandlung dienen. Dabei werden Ad-Hoc-Prozesse allerdings nicht, wie in der Literatur zum Teil der Fall [ReDa98, AaJa00, RiRD04, WRWR05], als temporäres oder dauerhaftes Verändern des Prozessschemas verstanden. Dadurch würde ein für das KMU-Umfeld zu komplexes Prozessmodell erforderlich sein. Stattdessen können vorübergehende informelle Wege innerhalb der Anwendung als alternativer, unstrukturierter Ablauf des Prozesses angeboten werden.

In zukünftiger Arbeit möchten wir außer der Integration von erweiterten Kommunikationsmitteln die vorgestellten Ansätze der Berechtigungsmodellierung mit einer einfachen Formulierung von Abläufen und Berechtigungen in einer *Domain-specific Language* (DSL) verbinden. Eine *Authorization Constraints*-DSL bildet dann den Rahmen für freiere Definitionen von Berechtigungen, wenn das erweiterte RBAC-Modell nicht ausreicht. Teil dieser Arbeit ist auch das Einbinden statischer und dynamischer Analysen der benötigten Berechtigungen für die einzelnen Abläufe und ob die Berechtigungen erfüllbar sind. Hier werden auch Erkenntnisse aus dem ORKA-Projekt¹ von Nutzen sein, in dem eine Autorisierungsarchitektur zur Modellierung von Berechtigungen entwickelt wird. Nicht zuletzt möchten wir weitere Erfahrungen mit den vorgestellten Ansätzen sammeln, um Nutzen und Risiken konkreter beziffern zu können.

¹ Siehe auch www.orka-projekt.de

Danksagung

Für dieses Dokument konnten wir wertvolle Erfahrungen innerhalb eines Projektes mit einem KMU sammeln. Wir bedanken uns für diese erfolgreiche Zusammenarbeit bei dem Unternehmen, das ungenannt bleiben möchte. Unser Dank geht auch an Dr. Karsten Sohr für seine inhaltlichen Hinweise und Ergänzungen aus dem Bereich des RBAC.

Literatur

- [AaJa00] W. M. P. van der Aalst, S. Jablonski: Dealing with workflow change: identification of issues and solutions. In: *International Journal of Computer Systems Science and Engineering*, 15, 5 (2000), 267–276.
- [AaWe05] W. M. P. van der Aalst, M. Weske: Case handling: a new paradigm for business process support. In: *Data Knowl. Eng.*, 53, 2 (2005), 129–162.
- [ASKP00] G.-J. Ahn, R. Sandhu, M. Kang, J. Park: Injecting RBAC to secure a Web-based workflow system. In: *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, ACM, New York, NY, USA (2000), 1–10.
- [AtHu96] V. Atluri, W. Kuang Huang: An Authorization Model for Workflows. In: *ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security*, Springer-Verlag, London, UK (1996), 44–64.
- [BeFA99] E. Bertino, E. Ferrari, V. Atluri: The specification and enforcement of authorization constraints in workflow management systems. In: *ACM Trans. Inf. Syst. Secur.*, 2, 1 (1999), 65–104.
- [CaCF01] F. Casati, S. Castano, M. Fugini: Managing Workflow Authorization Constraints through Active Database Technology. In: *Information Systems Frontiers*, 3, 3 (2001), 319–338.
- [CoLC04] D. Cohen, M. Lindvall, P. Costa: An introduction to agile methods. In: *Advances in Computers*, 62 (2004), 2–67.
- [FeKu92] D. Ferraiolo, R. Kuhn: Role-Based Access Controls. In: *15th NIST-NCSC National Computer Security Conference* (1992), 554–563.
- [GeHS95] D. Georgakopoulos, M. F. Hornick, A. P. Sheth: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. In: *Distributed and Parallel Databases*, 3, 2 (1995), 119–153.
- [HuKI98] P. Hung, K. Karlapalem, J. W. G. Iii: A Study of Least Privilege in CapBasED-AMS. In: *coopis*, 00 (1998), 208.
- [JAGS01] J. B. D. Joshi, W. G. Aref, A. Ghafoor, E. H. Spafford: Security models for web-based applications. In: *Commun. ACM*, 44, 2 (2001), 38–44.
- [LBBC⁺02] M. Lindvall, V. R. Basili, B. W. Boehm, P. Costa, K. Dangle, F. Shull, R. Tesoriero, L. A. Williams, M. V. Zelkowitz: Empirical Findings in Agile Methods. In: *Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe 2002*, Springer-Verlag, London, UK (2002), 197–207.

- [LiWL04] D.-R. Liu, M.-Y. Wu, S.-T. Lee: Role-based authorizations for workflow systems in support of task-based separation of duty. In: *J. Syst. Softw.*, 73, 3 (2004), 375–387.
- [MFWA⁺99] J. A. Miller, M. Fan, S. Wu, I. B. Arpinar, A. P. Sheth, K. J. Kochut: Security for the METEOR Workflow Management System. Tech. Rep., UGA-CS-LDIS, University of Georgia (1999).
- [OhPa03] S. Oh, S. Park: Task-role-based access control model. In: *Inf. Syst.*, 28, 6 (2003), 533–562.
- [PaSA01] J. S. Park, R. Sandhu, G.-J. Ahn: Role-based access control on the web. In: *ACM Trans. Inf. Syst. Secur.*, 4, 1 (2001), 37–71.
- [QuEP94] G. Quirchmayr, E. Ellmer, G. Pernul: Security for Workflow Management. In: *6th International Conference on Parallel and Distributed Computing and Systems*, Washington D.C., USA (1994).
- [ReDa98] M. Reichert, P. Dadam: Adeptflex: Supporting Dynamic Changes of Workflows Without Losing Control. In: *J. Intell. Inf. Syst.*, 10, 2 (1998), 93–129.
- [RiRD04] S. Rinderle, M. Reichert, P. Dadam: Flexible Support of Team Processes by Adaptive Workflow Systems. In: *Distrib. Parallel Databases*, 16, 1 (2004), 91–116.
- [R.J.01] B. R.A., E. J.H.P.: A framework for access control in workflow systems. In: *Information Management & Computer Security*, 9 (2001), 126–133.
- [SMLP05] Y. Sun, X. Meng, S. Liu, P. Pan: Flexible Workflow Incorporated with RBAC. In: *W. Shen, K.-M. Chao, Z. Lin, J.-P. A. Barthès, A. E. James (Hrsg.), CSCWD (Selected papers)*, Springer (2005), *Lecture Notes in Computer Science*, Bd. 3865, 525–534.
- [ThSa98] R. K. Thomas, R. S. Sandhu: Task-Based Authorization Controls (TBAC): A Family of Models for Active and Enterprise-Oriented Authorization Management. In: *Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Security XI*, Chapman & Hall, Ltd., London, UK, UK (1998), 166–181.
- [WaBK03] J. Wainer, P. Barthelmess, A. Kumar: W-RBAC - A Workflow Security Model Incorporating Controlled Overriding of Constraints. In: *Int. J. Cooperative Inf. Syst.*, 12, 4 (2003), 455–485.
- [WRWR05] B. Weber, M. Reichert, W. Wild, S. Rinderle: Balancing Flexibility and Security in Adaptive Process Management Systems. In: *R. Meersman, Z. Tari, M.-S. Hacid, J. Mylopoulos, B. Pernici, Ö. Babaoglu, H.-A. Jacobsen, J. P. Loyall, M. Kifer, S. Spaccapietra (Hrsg.), OTM Conferences (1)*, Springer (2005), *Lecture Notes in Computer Science*, Bd. 3760, 59–76.