Erschienen in: P. Horster (Hrsg.): Sichere Geschäftsprozesse, St. Leon-Rot, Sept. 2002, S. 30-42

Änderungsrechte in adaptiven Workflow-Management-Systemen

Manfred Reichert · Stefanie Rinderle

Universität Ulm
Abteilung Datenbanken und Informationssysteme
{reichert, rinderle}@informatik.uni-ulm.de

Zusammenfassung

Adaptive Workflow-Management-Systeme (WfMS) sind eine neuartige Technologie für die Realisierung flexibler, prozessorientierter Anwendungen. Sie gestatten es prozessbeteiligten Anwendern zur Laufzeit, in flexibler Art und Weise vom modellierten Ablauf abzuweichen, etwa durch Einfügen, Löschen oder Verschieben von Prozess-Schritten. Allerdings bieten adaptive WfMS derzeit noch keine ausreichenden Sicherheitskonzepte zur Kontrolle solcher Ad-hoc-Änderungen. Entsprechende Handlungsmöglichkeiten stehen entweder nur einzelnen Akteuren (z. B. Prozessverantwortlichen) offen, was vielfach zu restriktiv ist, oder aber sie können unkontrolliert durch beliebige Benutzer erfolgen. In diesem Beitrag diskutieren wir erstmals Anforderungen an Berechtigungskonzepte für Ad-hoc-Änderungen. Am Beispiel des Ad-hoc-Einfügens von Prozess-Schritten stellen wir systematisch dar, welche Berechtigungskonzepte konkret erforderlich sind und wie sich entsprechende Änderungsrechte möglichst einfach und kompakt definieren lassen. Dabei verfolgen wir einen rollenbasierten Ansatz, der in dem von uns entwickelten WfMS auch die Grundlage für die Definition und Verwaltung anderer Zugriffs- und Ausführungsrechte bildet. Unser Hauptaugenmerk gilt der komfortablen Festlegung und Pflege der Änderungsrechte. Entsprechende Berechtigungskonzepte bildet einen unverzichtbaren Bestandteil eines jeden adaptiven WfMS.

1 Einleitung

Die optimale Gestaltung ihrer Geschäftsprozesse gewinnt für Unternehmen zunehmend an Bedeutung. Deshalb müssen die funktions- und datenbezogenen Sichten in der betrieblichen Informationsverarbeitung konsequent um prozessorientierte Sichten erweitert werden. Erforderlich dazu sind vorgangsorientierte Informationssysteme, die Geschäftsprozesse in umfassender Weise ausführen, verwalten und überwachen können. Wichtig ist, dass sich entsprechende Anwendungen bei Bedarf rasch und fehlerfrei an geänderte Erfordernisse anpassen lassen.

Eine vielversprechende Technologie zur Realisierung prozessorientierter Informationssysteme bieten *Workflow-Management-Systeme* (WfMS) [Ober96, LeRo00]. Sie unterstützen die elektronische Abwicklung von Geschäftsprozessen entsprechend vordefinierter Arbeitsabläufe (engl. *Workflows*; kurz: WF). Zu diesem Zweck muss für jeden zu unterstützenden Prozesstyp vom Entwickler ein WF-Modell (*Prozessvorlage*) erstellt und im WfMS hinterlegt werden. Dazu bedient er sich in der Regel graphischer Beschreibungssprachen (z.B. Petri-Netze [Ober96, ElKR95]), die eine Festlegung der Modelle auf semantisch hoher Ebene gestatten. In die so erstellten WF-Modelle können Anwendungskomponenten eingesetzt werden, die zur Laufzeit bei der Ausführung einzelner

Prozess-Schritte (*Aktivitäten*) vom WfMS gerufen (und mit Daten versorgt) werden [LeRo00]. Soll nun ein neu auftretender Geschäftsfall elektronisch unterstützt werden, kann für ihn – ausgehend von der definierten Prozessvorlage – eine neue WF-Instanz erzeugt werden. Sie wird vom WfMS über ihre komplette Lebensdauer hinweg begleitet. Das WfMS übernimmt ihre Ausführung, Verwaltung und Überwachung, indem es die Durchführung einzelner Aktivitäten koordiniert, anstehende Aktivitäten für berechtigte Akteure in Arbeitslisten zusammenfasst und die zugehörigen Anwendungsprogramme (mit den richtigen Daten) bei der Ausführung dieser Aktivitäten ruft.

1.1 Problemstellung

Ein große Schwäche heutiger WfMS besteht darin, dass von dem einmal modellierten Ablauf zur Ausführungszeit nicht oder nur unter erheblichen Schwierigkeiten abgewichen werden kann. Eine solche starre Implementierung von Arbeitsprozessen stößt aber in vielen Anwendungsdomänen, etwa im Krankenhaus- und Engineering-Bereich, nur auf wenig Akzeptanz – selbst bei ansonsten idealer Prozessunterstützung [MüRa99, Reic00]. Hier muss es aufgrund der hohen Variabilität und Dynamik der Prozesse bei Bedarf möglich sein, in flexibler Form vom definierten Ablauf abzuweichen [DaRK00].

Aus diesem Grund werden in der einschlägigen WF-Literatur verschiedene Ansätze für adaptives WF-Management diskutiert [Aals99, ElKR95, EdLi98, Joer99, MüRa99, ReDa98, Teeg98, Wesk98]. Einen Schwerpunkt dieser Arbeiten bildet die adäquate Unterstützung spontaner Abweichungen vom modellierten Ablauf zur Laufzeit. Beispiele für Ad-hoc-Abweichungen sind das Einfügen, Löschen oder Verschieben von WF-Aktivitäten (vgl. Abb. 1). Wir bezeichnen im Folgenden ein WfMS, das für WF-Instanzen solche Laufzeitabweichungen zulässt, als *adaptiv*. Ebenso wie unsere eigenen Vorarbeiten zu adaptiven WfMS [ReDa98, BaRD01, DaRK00], haben sich bisherige Vorschläge aus diesem Bereich (z.B. [Aals99, Wesk98]) auf einige wenige Aspekte konzentriert, etwa den Erhalt von Konsistenzeigenschaften des WF nach seiner Abänderung oder auf die in diesem Zusammenhang notwendigen Zustandsanpassungen [ReDa98].

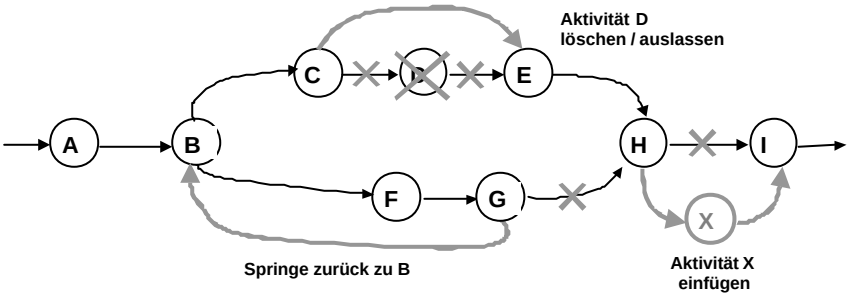

Abb. 1: Beispiele für Ad-hoc-Änderungen

Ebenfalls für die Praxis sehr wichtig, von bisherigen adaptiven WfMS aber noch nicht hinreichend berücksichtigt, sind Sicherheitsaspekte. Im Allgemeinen dürfen Ad-hoc-Änderungen einer WF-Instanz nicht unkontrolliert durch beliebige Personen erfolgen, sondern sollten nur ausgewählten Akteuren offenstehen. Ansonsten können unerwünschte Manipulationen nicht ausgeschlossen werden. Aber auch berechtigte Personen sollten davor geschützt werden, semantisch unsinnige Änderungen – soweit für das WfMS erkennbar – vorzunehmen. Aus diesem Grund werden für die Kontrolle von Ad-hoc-Änderungen, ebenso wie für die Erzeugung und Ausführung von WF-Instanzen, geeignete

Berechtigungskonzepte benötigt. In jedem Fall sollten Änderungsrechte – selbst bei großer Zahl von Benutzern und Aktivitätentypen – kompakt und verständlich definiert werden können. Um zu pflegbaren Rechtebasen zu gelangen, müssen sich entsprechende Rechtedefinitionen zudem robust gegenüber Änderungen zeigen, etwa in Bezug auf die Personalstruktur des Unternehmens (z. B. Wechsel eines Mitarbeiters zwischen zwei Bereichen). Adaptive WfMS haben bisher entweder stark vereinfachende Annahmen getroffen oder aber diese Fragestellungen gänzlich ausgeklammert. Auch Arbeiten, die sich speziell mit Sicherheitskonzepten in WfMS beschäftigen (z. B. [BeFA99]) haben bisher entsprechende Anforderungen nicht berücksichtigt.

1.2 Beitrag

In diesem Beitrag behandeln wir erstmals Berechtigungskonzepte zur Kontrolle von Ad-hoc-Änderungen in adaptiven WfMS. Wie sich solche Ad-hoc-Änderungen konkret festlegen lassen, was es dabei zu beachten gilt und welche Implementierungsfragestellungen sich ergeben, haben wir bereits in früheren Publikationen abgehandelt [ReDa98, DaRK00]. Hierauf gehen wird deshalb im Folgenden nicht mehr ein. An dieser Stelle sei lediglich erwähnt, dass bei dem von uns realisierten WfMS spontane Laufzeitabweichungen in der Folge niemals zu Inkonsistenzen oder Fehlern (z. B. Aufruf von Anwendungskomponenten mit fehlenden Daten, Blockierung der WF-Ausführung) führen [ReDa98]. Darüber hinaus können Ad-hoc-Änderungen, selbst bei großer Zahl von Benutzern und WF-Instanzen, effizient bewerkstelligt werden [BaRD01].

Aus Platzgründen beschränken wir uns im Folgenden auf Berechtigungskonzepte für *additive* Änderungen (Ad-hoc-Einfügung von Aktivitäten). Wir haben jedoch auch für die anderen von uns unterstützten Änderungsarten, etwa für das Löschen oder Verschieben von Aktivitäten, entsprechende Berechtigungskonzepte realisiert. Dabei bestehen keine fundamentalen Unterschiede im Vergleich zu den nachfolgend für das dynamische Einfügen vorgestellten Konzepte. Der Beitrag gliedert sich wie folgt: Abschnitt 2 fasst wichtige Grundlagen für die Definition von Ausführungs- und Änderungsrechten zusammen, die für das weitere Verständnis nötig sind. In Abschnitt 3 diskutieren wir, welche Art von Berechtigungen für das dynamische Einfügen von Aktivitäten auf WF-Instanzebene erforderlich sind. Wie sich entsprechende Berechtigungen kompakt definieren und verwalten lassen zeigt Abschnitt 4. Der Beitrag schließt mit einer kurzen Zusammenfassung und Diskussion.

2 Grundlagen

Zur Festlegung von Ausführungs- und Zugriffsberechtigungen dienen in WfMS *rollenbasierte* Zugriffskonzepte [FeKu92, FeCK95]. Um für WF-Aktivitäten zur Ausführungszeit potentielle Bearbeiter bestimmen zu können, müssen bei ihrer Definition (bzw. bei der Definition ihrer Vorlagen) geeignete Festlegungen getroffen werden. In der Regel werden dabei Aktivitäten keine konkreten Personen als Bearbeiter zugeordnet, sondern spezifische Ausführungsorgane, die bezüglich der Aufgabenzuteilung sog. *Rollen* übernehmen. Eine *Rolle* (z. B. Sachbearbeiter) fasst dabei eine Grundmenge von Fähigkeiten und Kompetenzen zusammen, welche entsprechende Benutzer zur Durchführung bestimmter Aktionen qualifizieren. Ausführungsorgane in derselben Rolle sind austauschbar, d. h. alle Personen, die sich über ihre Rolle für die Bearbeitung einer Tätigkeit

qualifizieren, können diese auch ausführen. Erforderlichenfalls kann ein Benutzer auch mehrere Rollen einnehmen.

In unserem Ansatz kann die Auswahl der potenziellen Bearbeiter einer Aktivität nicht nur auf Grundlage von Rollen, sondern auch in Abhängigkeit anderer Aspekte erfolgen, etwa der Zugehörigkeit einer Person zu einer bestimmten organisatorischen Einheit oder Arbeitsgruppe. Grundlage hierfür bilden die im *Organisationsmodell* hinterlegten Informationen. Das von uns verwendete Organisationsmetamodell zeigt Abb. 2, ein darauf basierendes Werkzeug für die Organisationsmodellierung ist in Abb. 3 dargestellt.

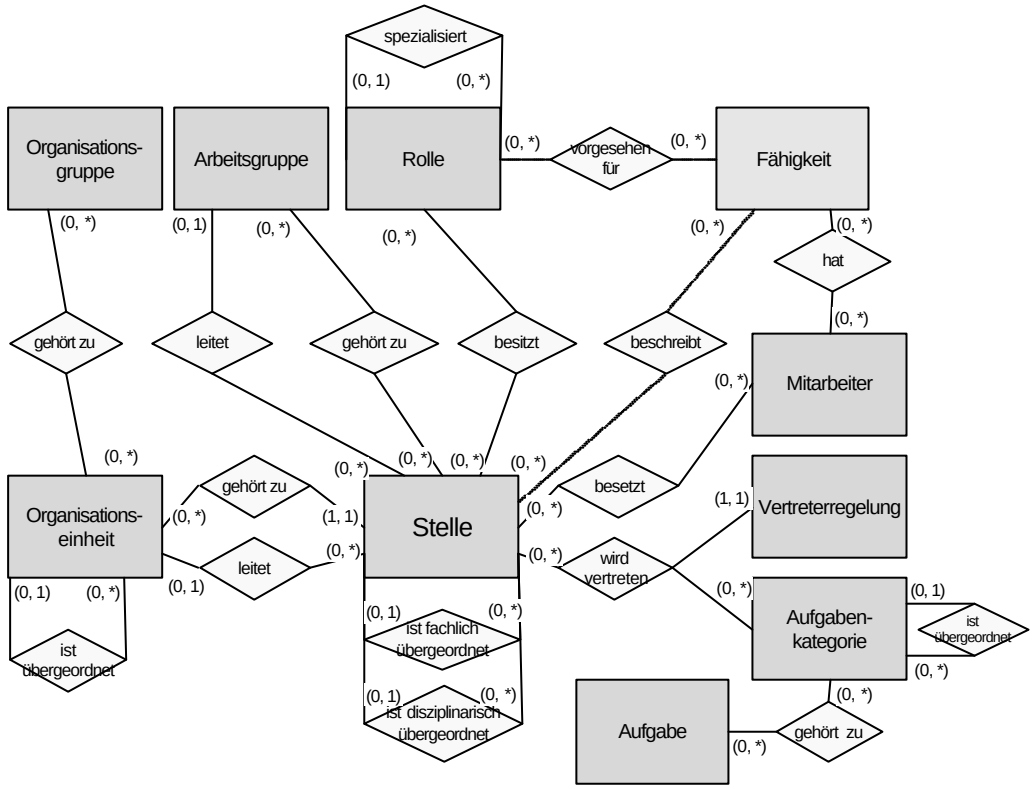

Abb. 2: Organisationsmetamodell

Ausgehend von einem konkreten Organisationsmodell können sog. *Bearbeiterzuordnungsausdrücke* (BZA) definiert und für die Festlegung von Ausführungs- und Zugriffsrechten verwendet werden. Einige einfache Beispiele für Bearbeiterzuordnungen zeigt Tabelle 1.

Wie in anderen WfMS auch, werden Bearbeiterzuordnungsausdrücke von uns verwendet, um für WF-Aktivitäten potentielle Bearbeiter festzulegen. Entsprechende Festlegungen können auch unabhängig von konkreten WF-Modellen vorgenommen werden, indem sie auf Typebene für wiederverwendbare Aktivitätenvorlagen erfolgen. Diese Vorlagen sind zusätzlich in Kategorien eingeteilt, die hierarchisch organisiert werden. Dadurch können Ausführungsrechte auch kompakt auf der Ebene von Kategorien festgelegt werden. Sie gelten dann auch für alle untergeordneten Kategorien und Vorlagen. Wir werden uns dieses einfache Prinzip auch bei der Definition von Änderungsrechten zunutze machen.

Tabelle 1: Beispiele für Bearbeiterzuordnungen

BZA	Bedeutung
Role = X	Benutzer hat Rolle X.
Role += X	Benutzer hat die angegebene oder eine speziellere Rolle
Unit = O	Benutzer gehört der Organisationseinheit O an.
Role = X and Unit = O	Benutzer hat die Rolle X und gehört der Organisationseinheit O an.
Capability =X	Benutzer hat Fähigkeit X
Position += X	Benutzer ist Vorgesetzter (beliebiger Ebene) der Stelle X
Unit-= O	Benutzer ist Mitglied einer Organisationseinheit, die O (auf beliebiger Ebene) untergeordnet ist
Actor = X	Benutzer ist der angegebene Mitarbeiter X

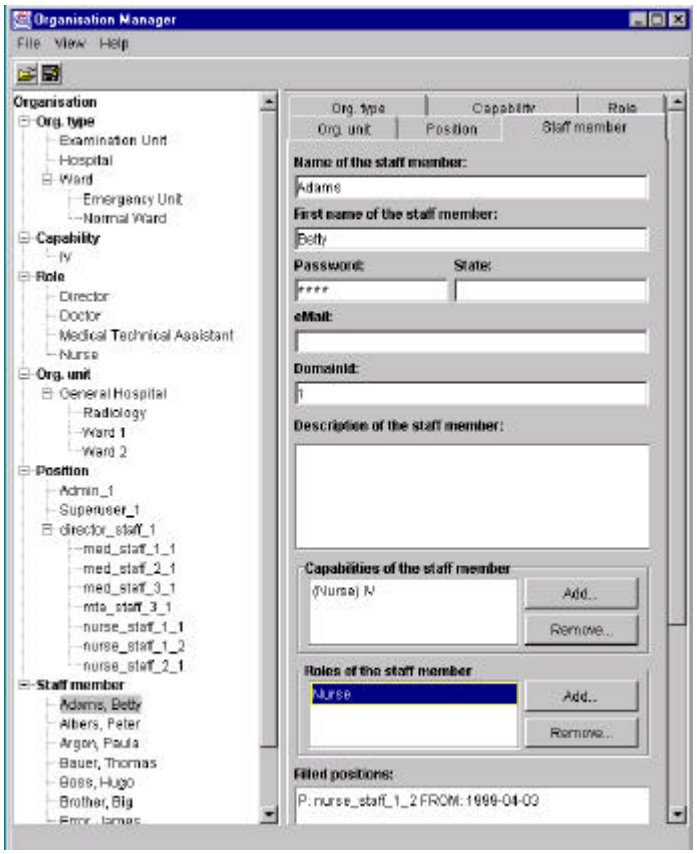

Abb. 3: Editor zur Erstellung von Organisationsmodellen

3 Erforderliche Rechte für Ad-hoc-Änderungen

In diesem Abschnitt zeigen wir am Beispiel des Ad-hoc-Einfügens von Aktivitäten (auf WF-Instanzebene), welche Berechtigungen hierzu erforderlich bzw. sinnvoll sind. Abschnitt 4 geht dann darauf ein, wie entsprechende Rechte definiert und verwaltet werden können.

1. **Benutzerabhängige Einschränkungen:** Auf welche Aktivitätentypen bzw. -vorlagen darf ein bestimmter Benutzer beim Einfügen von WF-Schritten Bezug nehmen?

   Diese Frage stellt sich unabhängig vom Typ der WF-Instanz, in die eine neue Aktivität eingefügt werden soll. Hintergrund ist, dass einem Benutzer im Allgemeinen nicht alle im Repository hinterlegten Aktivitätenvorlagen zur Auswahl stehen. Ein Verwaltungsangestellter eines Krankenhauses etwa darf (mangels Kompetenz) keine medizinische oder pflegerische Tätigkeit in den laufenden Behandlungsprozess eines Patienten einfügen. Die Menge der Aktivitätentypen bzw. -vorlagen, auf die ein bestimmter Benutzer beim Einfügen neuer WF-Schritte Bezug nehmen darf, muss deshalb a priori sinnvoll eingeschränkt werden.

2. **Prozesstypabhängige Einschränkungen:** Aktivitäten welchen Typs bzw. welcher Vorlage dürfen in WF-Instanzen eines bestimmten WF-Typs eingefügt werden?

   Diese Frage stellt sich unabhängig davon, von welcher Person die Einfügung vorgenommen wird. Im Allgemeinen sollten nur solche Einfügungen gestattet werden, die semantisch sinnvoll sind. Deshalb dürfen zu den WF-Instanzen eines bestimmten WF-Typs nicht beliebige Aktivitäten hinzugefügt werden. Wir wollen dies durch ein einfaches Beispiel illustrieren: Wurde zur Sammelbestellung von Medikamenten von einer Krankenstation ein Bestellvorgang gestartet, der unabhängig von der Behandlung eines konkreten Patienten ist, macht das Einfügen patientenbezogener Aktivitäten (z.B. „Patient waschen“) in diese WF-Instanz semantisch keinen Sinn und ist deshalb zu unterbinden. Aus der Tatsache, dass ein Benutzer eine bestimmte Aktivität in die Instanzen eines WF-Typs X einfügen darf, kann also im Allgemeinen nicht geschlossen werden, dass er dieselbe Aktivität auch in WF-Instanzen anderer WF-Typen einfügen darf.

3. **Prozessinstanzabhängige Einschränkungen:** Welche Benutzer dürfen eine bestimmte Aktivität in eine gegebene WF-Instanz einfügen?

   Die Einschränkungen 1 und 2 reichen allein noch nicht aus, um entscheiden zu können, ob ein Benutzer eine bestimmte Aktivität in eine WF-Instanz einfügen darf oder nicht. Im Allgemeinen kann es zu einem WF-Typen zugehörige WF-Instanzen geben, für die ein Benutzer eine Aktivität X einfügen darf, wohingegen ihm dies für andere WF-Instanzen desselben WF-Typs nicht gestattet ist. Beispielsweise können radiologische Untersuchungen von unterschiedlichen Stationen im Krankenhaus angefordert werden. Generell darf ein Stationsarzt einen zusätzlichen Vorbereitungsschritt in die WF-Instanz eines solchen Untersuchungsablaufs nur dann einfügen, wenn er der behandelnden Einheit (d.h. Station) des Patienten angehört. Das bedeutet, dass Berechtigungskonzepte für das Ad-hoc-Einfügen von Aktivitäten variabel gestaltet werden können müssen. D.h. sie sind abhängig von gewissen WF-Attributen, die erst zur Laufzeit mit einem Wert belegt werden.

4. **Kontextabhängige Einschränkungen:** In welche Bereiche des WF-Graphen einer WF-Instanz dürfen Aktivitäten eingefügt werden?

   Die bisherigen Festlegungen regeln, wer welche Art von Aktivitäten in welche WF-Instanzen einfügen darf. Nicht festgelegt ist jedoch, an welchen Positionen des WF-Graphen entsprechende Einfügungen erfolgen können sollen.

Die Fragestellungen 1 – 3 werden nachfolgend behandelt. Den zuletzt genannten Aspekt klammern wir in diesem Beitrag jedoch aus. Hierzu wurden von uns ebenfalls Lösungskonzepte erarbeitet. Sie machen sich zum einen die hierarchische Strukturierung von WF-Modellen zunutze, zum anderen erlauben sie es, WF-Graphen in sog. Änderungssphären zu unterteilen. Abschließend sei erwähnt, dass der Akteur, der eine Ad-hoc-Einfügung vornimmt, nicht notwendigerweise dieselbe Person sein muss, die anschließend den neu eingefügten Schritt bearbeitet. In unserem Ansatz trennen wir strikt zwischen Ausführungs- und Einfügerechten. In diesem Beitrag konzentrieren wir uns auf Letztere.

4 Definition und Verwaltung von Änderungsrechten

In diesem Abschnitt gehen wir darauf ein, wie die vorangehend diskutierten Einschränkungen gewährleistet werden können. Dazu zeigen wir, wie Änderungsrechte im Allgemeinen zu definieren und verwalten sind, unter Beachtung der eingangs genannten Anforderungen (z.B. Pflegbarkeit der Rechtebasis).

4.1 Benutzerabhängige Einschränkungen

Für Benutzer muss kompakt beschreibbar sein, auf welche Aktivitätentypen bzw. –vorlagen sie beim Einfügen neuer WF-Schritte Bezug nehmen dürfen. Entsprechende Festlegungen müssen im Allgemeinen unabhängig von konkreten WF-Typen oder –Instanzen erfolgen können. Dadurch kann zum einen die Menge der von einer bestimmten Benutzergruppe einfügbaren Aktivitäten a priori sinnvoll eingeschränkt werden, zum anderen ergibt sich eine (pflegbare) Rechtebasis, die sich robust gegenüber Prozessvorlagen-Änderungen zeigt.

Prinzipiell können solche Einfügerechte auf Grundlage der erwähnten Bearbeiterzuordnungsausdrücke (BZA) festgelegt werden. Ein naiver Ansatz wäre es allerdings, für jeden einzelnen Aktivitätentyp eine entsprechende Berechtigung (in Form eines BZA) zu definieren. Dies wäre bei großer Zahl von Aktivitätenvorlagen mit einem zu hohen Aufwand für die Rechtedefinition und -pflege verbunden. Aus diesem Grund unterstützen wir ein hierarchisches Rechtemodell, bei dem Berechtigungen nicht nur für einzelne Vorlagen, sondern – wo sinnvoll und möglich – auch für komplette Aktivitätenkategorien vereinbart werden können. Der für eine bestimmte Aktivitätenkategorie festgelegte BZA ist dann auch für alle untergeordneten Aktivitätenkategorien und -vorlagen wirksam, es sei denn, er wird dort explizit durch einen anderen BZA überschrieben. In der Mehrzahl der Fälle wird es genügen, entsprechende Berechtigungen auf der Ebene von Aktivitätenkategorien zu definieren.

Wir wollen dies anhand eines Beispiels illustrieren. Dazu seien der Aktivitätenbaum aus Abb. 4 und die ihm zugeordneten Berechtigungen gegeben. Die Rechte für das Einfügen von Aktivitäten sind hier auf Ebene der Aktivitätenkategorien A_1, A_2 und A_3 festgelegt. Sie gelten somit auch für alle untergeordneten Aktivitätenkategorien und -vorlagen. Beispielsweise ist die Berechtigung *Role = Arzt* für

alle untergeordneten Aktivitätenvorlagen von A_1 wirksam. Demnach darf ein Benutzer mit dieser Rolle beim Einfügen von WF-Schritten auf die Aktivitätenvorlagen a_{11}, a_{12}, a_{13}, ... Bezug nehmen, es sei denn diese Menge wird durch andere Einschränkungen (siehe unten) weiter verkleinert. Abschließend sei erwähnt, dass keinerlei Anpassung erforderlich wird, wenn sich in Bezug auf die Zusammensetzung dieser Aktivitätenkategorien Änderungen ergeben (z.B. durch Hinzunahme neuer Aktivitätenvorlagen). Dadurch reduziert sich der Aufwand für die Definition und Pflege der Einfügerechte erheblich.

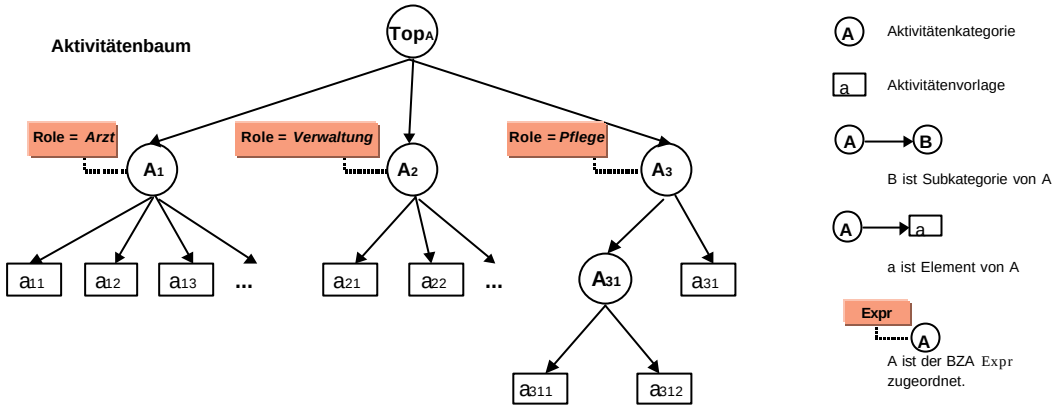

Abb. 4: Aktivitätenbaum mit Bearbeiterzuordnungen

In Abb. 4 sind Änderungsrechte nur für die Aktivitätenkategorien der obersten Stufe definiert und gelten somit jeweils für alle untergeordneten Aktivitätenkategorien und -vorlagen. Wie erwähnt, ist es aber auch möglich, diese Rechte auf untergeordneter Ebene bei Bedarf zu überschreiben. Wir illustrieren dies anhand zweier Beispiele:

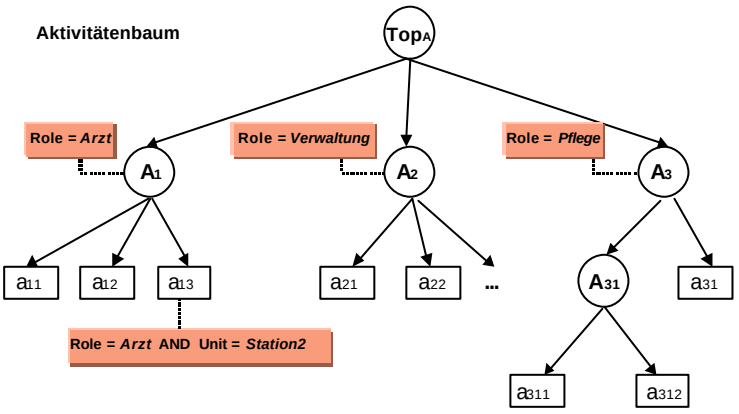

Abb. 5: Rechte überschreiben (Verkleinerung der Menge berechtigter Personen)

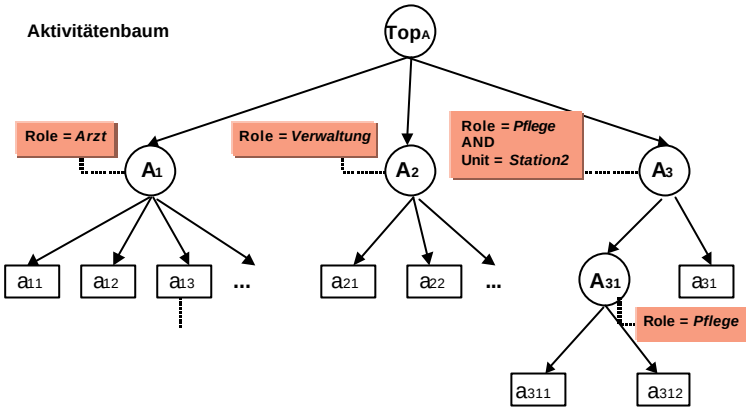

Abb.6: Rechte überschreiben (Vergrößerung der Menge berechtigter Personen)

In Abb. 5 ist der für A_1 festgelegte BZA *Role = Arzt* nur für a_{11} und a_{12} wirksam, nicht jedoch für die Aktivitätenvorlage a_{13}. Für sie wurde der auf der Ebene von A_1 festgelegte BZA durch den BZA *Role = Arzt AND Unit = Station 2* überschrieben. Demnach dürfen Aktivitäten dieses Typs nur von Benutzern mit Rolle *Arzt* und mit Zugehörigkeit zur Organisationseinheit *Station2* eingefügt werden. Die Menge der zum Einfügen von a_{13} berechtigten Akteure wird somit gegenüber A_1 (bzw. a_{11} und a_{12}) verkleinert. Ein zweites Beispiel für das Überschreiben von Einfügerechten zeigt Abb. 6. Hier ist die für die Kategorie A_3 festgelegte Berechtigung *Role = Pflege AND Unit = Station2* für die Aktivitätenvorlage a_{31} wirksam, nicht jedoch für die Elemente der untergeordneten Kategorie A_{31}. Im Gegensatz zum vorangehenden Beispiel aber vergrößert hier der BZA *Role = Pflege* die Menge berechtigter Akteure.

Soll nun für einen konkreten Benutzer festgestellt werden, auf welche Aktivitätenvorlagen er sich beim Einfügen von Aktivitäten in WF-Instanzen generell beziehen darf, muss der Aktivitätenbaum (teilweise) traversiert werden. Vermerkt man dabei für Nichtblattknoten, ob es auf untergeordneter Ebene zu einem Überschreiben von Rechtefestlegungen kommt oder nicht, kann der Aufwand für diese Überprüfung minimal gestaltet werden.

4.2 Prozesstypabhängige Einschränkungen

Wie motiviert, muss für jeden WF-Typ geregelt sein, welche Aktivitäten (bzw. Aktivitäten welchen Typs) in zugehörige WF-Instanzen eingefügt werden dürfen und welche nicht. Dies sollte unabhängig von den Benutzern, die derartige Einfügungen vornehmen, erfolgen können. Ferner ist zu beachten, dass solche prozesstypbezogenen Berechtigungen ebenfalls wieder mit vertretbarem Aufwand definiert und gepflegt werden können. Im Idealfall sollten entsprechende Rechte nicht angepasst werden müssen, wenn neue WF-Typen hinzukommen oder bestehende WF-Typen wegfallen.

Bezugspunkt für die Definition entsprechender Rechte bilden die im Repository hinterlegten Prozesskategorien und -vorlagen. Sie sind – ebenso wie Aktivitätenkategorien und -vorlagen – hierarchisch organisiert, wobei Prozesskategorien durch Nichtblattknoten und Prozessvorlagen durch Blattknoten im *Prozessbaum* repräsentiert werden. Den Knoten dieses Baums wird nun direkt oder indirekt eine Menge von Aktivitätenvorlagen zugeordnet, auf die beim Einfügen neuer Schritte Bezug genommen werden darf. Bei ihrer Festlegung können Aktivitätenvorlagen und -kategorien referenziert werden, wodurch wieder eine kompakte Rechtedefinition möglich wird.

Die einer bestimmten Prozesskategorie zugeordnete Aktivitätenmenge ist auch für alle untergeordneten Prozesskategorien und -vorlagen gültig, kann dort aber noch um zusätzliche Aktivitäten erweitert werden. D.h. im Gegensatz zu dem in Abschnitt 4.1 beschriebenen Vorgehen werden Rechtefestlegungen auf untergeordneter Ebene nicht überschrieben, sondern können dort um weitere Rechte ergänzt werden (additive Semantik). Ein einfaches Beispiel zeigt Abb. 7. Hier wird der obersten Prozesskategorie Top_P die Aktivitätenvorlage a_{21} zugeordnet. Diese Festlegung gilt dementsprechend auch für alle untergeordneten Prozesskategorien und -vorlagen. Das bedeutet, dass Aktivitäten mit Vorlage a_{21} in Instanzen beliebiger Prozessvorlagen eingefügt werden dürfen, es sei denn dies wird durch andere (orthogonale) Einschränkungen (vgl. Abschnitte 4.1 und 4.3) unterbunden. Auf untergeordneter Ebene erfolgt noch eine Vergrößerung der Menge einfügbarer Aktivitäten. Beispielsweise können Aktivitäten der Kategorie A_2, aufgrund der Zuordnung von A_2 zu P_1, in alle untergeordneten Prozessvorlagen von P_1 (bzw. in deren Instanzen) eingefügt werden (p_{111}, p_{112} und p_{121}). Für Instanzen der Prozessvorlagen p_{111} und p_{112} können zusätzlich noch Aktivitätenvorlagen der Kategorie A_1 hinzukommen (wg. der Zuordnung von A_1 zu P_{11}).

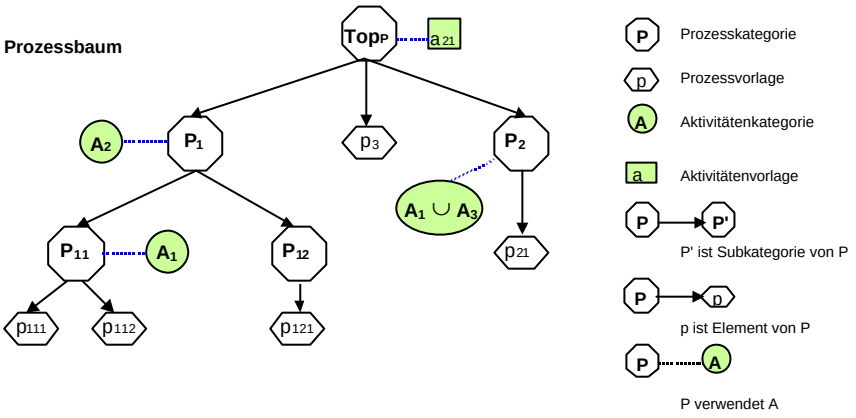

Abb. 7: Prozessbaum mit zugeordneten Aktivitätenkategorien und -vorlagen

Mit dem vorgestellten Konzept können Aktivitätenvorlagen den Prozesskategorien bzw. -vorlagen übersichtlich und kompakt zugeordnet werden. Verzichtet man dabei auf die Zuordnung einzelner Aktivitätenvorlagen und referenziert stattdessen Aktivitätenkategorien, so müssen im Prozessbaum keine Anpassungen erfolgen, wenn später neue Aktivitätenvorlagen hinzukommen oder bestehende Aktivitätenvorlagen wegfallen. Ähnliches gilt für den Fall, dass neue Prozessvorlagen in den Prozessbaum eingefügt werden sollen. Hier können im Regelfall die Festlegungen der übergeordneten Prozesskategorien übernommen werden.

Zur Ermittlung der Aktivitätenvorlagen, auf die ein Benutzer u beim Einfügen eines Schrittes in eine Prozessinstanz (mit Vorlage p) Bezug nehmen darf, kann wie folgt vorgegangen werden:

1. Bestimmung aller Aktivitätenvorlagen, die von dem Benutzer u generell eingefügt werden dürfen, unabhängig vom Typ der betrachteten WF-Instanz (→ *UserActivitySet*). Grundlegend hierfür sind die in Abschnitt 4.1 skizzierten Konzepte sowie das für den Benutzer u hinterlegte Profil (z.B. Rollen von u und zugeordnete Organisationseinheit).

2. Bestimmung aller Aktivitätenvorlagen, die in Instanzen der Prozessvorlage p eingefügt werden dürfen, unabhängig von den Personen die diese Änderung vornehmen (→ *ProcessType-ActivitySet*).[1]
3. Der Benutzer darf dann beim Einfügen neuer Schritte in die Prozessinstanz auf Aktivitätenvorlagen der Menge *UserActivitySet* ∩ *ProcessTypeActivitySet* Bezug nehmen.

Der in Abb. 7 dargestellte Sachverhalt kann invertiert werden, indem man im Aktivitätenbaum den einzelnen Knoten die entsprechenden Prozesskategorien oder -vorlagen zuweist. Dies bietet zur Beantwortung gewisser Fragestellungen Vorteile, auf deren Darstellung wir an dieser Stelle aber aus Platzgründen verzichten.

4.3 Prozessinstanzabhängige Einschränkungen

Wie in Kapitel 3 diskutiert, reichen die Berechtigungskonzepte aus den Abschnitten 4.1 und 4.2 allein noch nicht aus, um ermitteln zu können, ob ein Benutzer eine bestimmte Aktivitätenvorlage in eine gegebene WF-Instanz auch wirklich einfügen darf oder nicht. Der Grund hierfür ist, dass ein und dieselbe Prozessvorlage im Allgemeinen in unterschiedlichen Unternehmensbereichen instanziiert werden kann. Dementsprechend variabel müssen die Berechtigungen für die Änderung entsprechender WF-Instanzen gestaltet werden.

Die Rechtedefinition kann ähnlich wie in den beiden vorangehenden Fällen erfolgen, indem man den Knoten des Prozessbaums *variable Bearbeiterzuordnungsausdrücke* zuweist. Sie regeln, welche WF-Instanzen eines bestimmten WF-Typs bzw. einer bestimmten Prozesskategorie von einem Benutzer manipuliert werden dürfen. Wir wollen dies anhand eines einfachen Beispiels illustrieren: In Abb. 8 ist der Prozesskategorie P_1 der variable Bearbeiterzuordnungsausdruck *Role = Arzt AND Unit = Attr(_BEHANDELNDE_STATION)* zugewiesen. Diese Festlegung gilt sowohl für P_1 als auch für die untergeordneten Prozesskategorien und -vorlagen P_{11}, P_{12}, p_{111} und p_{121}. Sie ist nicht für p_{112} gültig, da hier ein speziellerer BZA festgelegt wurde.

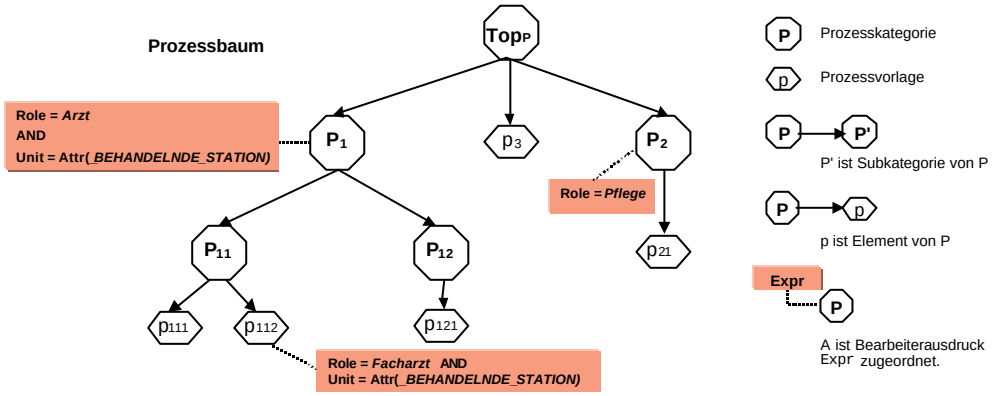

Abb. 8: Prozessbaum mit Bearbeiterzuordnungen

[1] Diese Menge kann bestimmt werden, indem man im Prozessbaum den Weg von der Wurzel zu dem Blatt mit der entsprechenden Prozessvorlage p durchläuft und dabei für jeden Knoten die zugeordneten Aktivitätenmengen „aufaddiert". Bezogen auf das Beispiel aus Abb. 7 ergibt sich so für p_{112} die Aktivitätenmenge $\{a_{21}\} \cup A_2 \cup A_1$.

Zu beachten ist, dass der für den Knoten P_1 festgelegte BZA variabel ist. Ein Benutzer darf demnach in entsprechende WF-Instanzen nur dann Schritte einfügen, wenn er die Rolle *Arzt* besitzt und wenn er der Behandlungseinheit *_BEHANDELNDE_STATION* angehört. Letzteres ist ein für Prozessvorlagen der Kategorie P_1 (sowie untergeordneter Kategorien) definiertes Attribut, dessen konkreter Wert erst bei Erzeugung der WF-Instanzen festgelegt wird. Dadurch wird es möglich, Einfügerechte in Abhängigkeit von Attributen der WF-Instanzen, variabel zu gestalten.

5 Zusammenfassung und Ausblick

In diesem Beitrag haben wir ein Sicherheitskonzept vorgestellt, das eine flexible Kontrolle von Ad-hoc-Abweichungen bei der WF-Ausführung ermöglicht. Solche spontanen WF-Änderungen können erforderlich werden, um flexibel auf Ausnahmen oder geänderte Realweltsituationen reagieren zu können. Sie dürfen aber niemals unkontrolliert erfolgen, da ansonsten ein Missbrauch nicht ausgeschlossen werden kann.

In diesem Beitrag haben wir erstmals wichtige Anforderungen, die in diesem Zusammenhang bestehen, erörtert und am Beispiel des Ad-hoc-Einfügens von WF-Schritten aufgezeigt wie entsprechende Berechtigungskonzepte definiert und verwaltet werden können. Unser Hauptaugenmerk galt dabei der möglichst einfachen Festlegung und Pflegbarkeit der Änderungsrechte. Unsere konkreten Erfahrungen mit Prozessen aus dem Krankenhausbereich haben gezeigt, dass diese beiden Ziele mit den vorgestellten Konzepten erreicht werden können. Grundlegend ist die hierarchische Strukturierung von Prozess- und Aktivitätenkategorien sowie die Verwendung (erweiterter) rollenbasierter Zugriffskonzepte. Im Wesentlichen haben wir uns also an Konzepten orientiert, wie sie in gängigen WfMS sowie in dem von uns entwickelten ADEPT-WfMS vorzufinden sind.

Interessante Fragestellungen ergeben sich noch im Zusammenhang mit unternehmensübergreifenden Prozessen. Hier haben Anwender oftmals keine globale Sicht auf den Prozess, so dass Änderungen in der Regel immer nur Fragmente eines WF-Modells betreffen können. Abgesehen davon ist es im Allgemeinen auch nicht tolerabel, dass eine Person eines Unternehmens A in die (Teil-)Abläufe eines Unternehmens B eingreift bzw. diese ändert. Ebenfalls nicht behandelt haben wir die Fragestellung, inwieweit auch andere Benutzer bei der Definition einer Änderung einbezogen werden müssen (etwa um der beabsichtigten Modifikation zuzustimmen).

Literatur

[Aals99] v. der Aalst, W.: Generic Workflow Models: How to Handle Dynamic Change and Capture Management Information? Proc. CoopIS'99, Edinburgh, 1999, S. 115–126

[BeFA99] Bertino, E.; Ferrari, E.; Atluri, V.: The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. ACM Transactions on Information and System Security, 2(1):65-104, Februar 1999

[BaRD01] Bauer, Th.; Reichert, M.; Dadam, P.: Adaptives und verteiltes Workflow-Management. Proc. BTW'2001, S. 47-66, Oldenburg, März 2001

[DaRK00] Dadam, P.; Reichert, M.; Kuhn, K.: Clinical Workflows - The Killer Application for Process-oriented Information Systems? Proc. BIS‘2000, Posen, 2000, S. 36–59

[ElKR95] Ellis, C.A.; Keddara, K.; Rozenberg, G.: Dynamic Change within Workflow Systems. Proc. Conf. Org. Computing Systems, Milpitas, CA, S. 10–21, 1995

[EdLi98] Eder, J.; Liebhart, W.: Contributions to Exception Handling in Workflow-Management. Proc. EDBT-Workshop on Worklfow Management Systems, Valencia, 1998, S. 3-10

[FeKu92] Ferraiolo, D.F.; Kuhn, D.R.: Role-based Access Control. Proc. 15th National Computer Security Conference, 1992

[FeCK95] Ferraiolo, D.F.; Cugini, J.A.; Kuhn, D.R.: Role-based Access Control (RBAC) - Features and Motivations. Proc. 11th Conf. Computer Security Applications, November 1995

[Joer99] Joeris, G.: Defining Flexible Workflow Execution Behaviors. Proc. Workshop on Enterprise-wide and Cross-Enterprise Workflow-Mgmt., Paderborn, 1999, S. 49–55

[LeRo00] Leymann, F.; Roller, D.: Production Workflow. Prentice Hall, 2000.

[MüRa99] Müller, R.; Rahm, E.: Rule-Based Dynamic Modification of Workflows in a Medical Domain. Proc. BTW '99, Freiburg, 1999, S. 429-448

[Ober96] Oberweis, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen. Teubner, 1996

[ReDa98] Reichert, M.; Dadam, P.: $ADEPT_{flex}$ – Supporting Dynamic Changes of Workflows Without Losing Control. Journal of Intelligent Inf. Systems, 10(2):93-129, 1998

[Reic00] Reichert, M.: Prozessmanagement im Krankenhaus - Nutzen, Anforderungen und Visionen. das Krankenhaus, 92(11):903-909, November 2000

[Teeg98] Teege, G.: Flexible Workflows. Proc. Workshop Flexibilität und Kooperation in Workflow-Management-Systemen, 1998, S. 13–21.

[Wesk98] Weske, M.: Flexible Modeling and Execution of Workflow Activities. Proc. 31st Hawaii Int'l Conf. on Sys. Sciences, Software Technology Track, 1998, S. 713–722