Gesellschaft für Informatik e.V. (GI) GI. (Hrsg.): INFORMATIK 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2021 1503

Effiziente Modellierung von Eskalationen und Stellvertretungen in Geschäftsprozessen

Thomas Bauer[1]

Abstract: Eskalationen und Stellvertretungen sind wichtig, um bei der Ausführung von Geschäftsprozessen (GP) auf Ausnahmesituationen reagieren zu können. Hierbei erfordert die Festlegung der jeweils gewünschten Zielperson(en) ähnlich komplexe Regeln, wie die Zuordnung von Aktivitäten zu regulären Bearbeitern. Zusätzlich sind unterschiedliche Regeln für verschiedene Aktivitäten, Originalbearbeiter, etc. erforderlich. Sehr viele solche Regeln zu definieren, würde aber einen großen Aufwand für die GP-Designer bedeuten. Deswegen sollte eine Regel für mehrere Aktivitäten verwendbar sein. Dies wird möglich, indem sie sich auf die reguläre Bearbeiterzuordnung der betroffenen Aktivität beziehen kann, z.B. auf die dort festgelegte Rolle. Da in einer Bearbeiterzuordnung mehrere Rollen gefordert sein können, muss zusätzlich festgelegt werden, welche von diesen referenziert werden soll. Deshalb führen wir für Objekte des Organisationsmodells einen benutzerdefinierbaren SubType ein. Dieser kann bei der Spezifikation der Zielpersonen einer Eskalation oder Stellvertretung verwendet werden, um die gewünschte Rolle auszuwählen. Dadurch wird eine solche Regel für viele unterschiedliche Aktivitäten verwendbar.

Keywords: Geschäftsprozess, Organisatorische Perspektive, Eskalation, Stellvertretung.

1 Motivation

Die organisatorische Perspektive von Prozess Management Systemen (PMS) definiert, welche Personen eine bestimmte Aktivität in einem Geschäftsprozess (GP) bearbeiten sollen (mittels einer sog. Bearbeiterzuordnung) und auch, wie sich das PMS in Ausnahmesituationen verhalten soll. [BL20] stellt hierzu Ansätze aus der Wissenschaft vor, und zeigt, dass heutzutage einige Aspekte in kommerziellen PMS nur unzureichend umgesetzt sind. So ist es i.d.R. nicht möglich, mächtige Bearbeiterzuordnungen auf einfache (z.B. graphische) Art und Weise zu definieren.

Startet oder beendet ein Bearbeiter eine Aktivität nicht innerhalb der vorgegebenen Frist, so wird eine *Eskalation*[2] ausgelöst. Dann wird z.B. ein Vorgesetzter informiert oder die Aktivität wird einem anderen Benutzer zugeordnet. Ist ein Bearbeiter längere Zeit abwesend (z.B. in Urlaub), so legt eine *Stellvertreterregelung* fest, wer diese Aktivität stattdessen bearbeiten soll. Eskalationen und Stellvertretungen haben gemeinsam, dass zusätzliche Personen involviert werden sollen. Diese Zielpersonen müssen in beiden Fällen abhängig von dem Typ der betroffenen Aktivität, dem Originalbearbeiter und dem

[1] Hochschule Neu-Ulm, Wileystr. 1, 89231 Neu-Ulm, thomas.bauer@hnu.de

[2] Im Folgenden ist mit Eskalation ausschließlich ein solches Verhalten gemeint, d.h. Aktionen, die ausgelöst werden, wenn der Start oder die Beendigung einer Aktivität des GP nicht rechtzeitig erfolgt.

CC BY SA

Prozesskontext definiert werden können. Deshalb bestehen für beide Themen ähnliche Anforderungen. In [BL20] wurde aufgezeigt, dass für die Modellierung solcher Zielpersonen, aktuell keine geeigneten wissenschaftlichen Konzepte existieren, und dass Eskalationen und Stellvertretungen in kommerziellen PMS heutzutage oft nur rudimentär umgesetzt sind. Deshalb beschäftigt sich dieser Beitrag mit dieser Problemstellung.

Das Problem hierbei ist, dass die Festlegung der Regeln zur Spezifikation der Zielpersonen (Target Person Rule: TPR) ohne einen zu großen Aufwand für die GP-Designer erfolgen soll (d.h. effizient). Das ist nur erreichbar, wenn möglichst wenige solche Regeln erforderlich sind. Hierzu muss eine einzelne TPR für viele unterschiedliche Originalbearbeiter und Aktivitäten gültig sein. Dies ist z.B. sehr einfach möglich, wenn die Zielpersonen einer Eskalation oder Stellvertretung mittels einer festen Rolle definiert werden können (z.B. „*Role = Arzt*“). Es sollen aber auch komplexe TPR verwendbar sein, wie z.B. „*Role = HW-Entwickler ∧ selbe Projektzugehörigkeit wie der Originalbearbeiter*“ (hierbei soll das Organisationsmodellobjekt Projekt eine Gruppe sein, z.B. *Entwicklung C-Klasse*, *Entwicklung S-Klasse*). Damit diese TPR für viele Aktivitäten und Bearbeiter unterschiedlicher Projekte (in identischer Form) verwendet werden kann, muss sie sich auf die „normale“ Bearbeiterzuordnung der betroffenen Aktivität beziehen können, weil ausschließlich diese auf das (jeweils unterschiedliche) Projekt verweist.

Eine solche Referenzierung soll natürlich nicht nur für Projekte (also Gruppen) möglich sein, sondern für alle Objekttypen des Organisationsmodells: Im Folgenden wird von den Objekttypen Rolle, Gruppe, Kompetenz und Organisationseinheit (letztere inkl. einer Hierarchie) ausgegangen, wobei eine Erweiterung leicht möglich ist (s. Abschnitt 3.1). Verallgemeinert man die oben dargestellte TPR zu „*selbe Rolle und selbe Gruppe wie die Originalaktivität*“, dann kann diese zusätzlich Regel z.B. auch für Aktivitäten verwendet werden, in deren Bearbeiterzuordnung eine andere Rolle gefordert ist, wie z.B. „*Role = Testfahrer ∧ selbe Projektzugehörigkeit wie der Originalbearbeiter*“. Die Notwendigkeit zur Definition einer weiteren TPR entfällt also. Schließlich können in einer Bearbeiterzuordnung auch mehrere Rollenzugehörigkeiten gefordert sein, von denen nur manche für die TPR relevant sind, z.B. „*Role = HW-Entwickler ∧ Role = Senior Entwickler ∧ selbe Projektzugehörigkeit wie Vorgängeraktivität X*“. Für die TPR soll in diesem Beispiel die zweite Rollenzuordnung nicht berücksichtigt werden. Deshalb muss irgendwie ausgedrückt werden, welche der beiden Rollenzugehörigkeiten relevant ist. Die direkte Verwendung des Rollennamens (hier: *HW-Entwickler*) verbietet sich, weil eine so eng gefasste Regel dann nicht mehr für andere Aktivitäten verwendet werden könnte, z.B. mit Bearbeiterzuordnung „*Role = Testfahrer*“. Ebenso wäre eine solche Regel nicht bei Bearbeiterzuordnungen verwendbar, bei denen sich der Rollenname dynamisch ergibt, z.B. aus einer Prozessvariablen (d.h. Anwendungsdaten).

Um dieses Problem zu lösen, müssen unterschiedliche Arten von Organisationsmodellobjekten (z.B. unterschiedliche Arten von Rollen) unterscheidbar gemacht werden. Hierzu führen wir sog. SubTypes ein. Jedem Organisationsmodellobjekt (d.h. jeder Rolle, jeder Gruppe, etc.) kann ein SubType zugeordnet werden. So wird z.B. der Rolle *HW-Entwickler* der SubType *Tätigkeit* und der Rolle *Senior Entwickler* der SubType

Skill-Level zugeordnet. Die Gruppen *Entwicklung C-Klasse* und *Entwicklung S-Klasse* sind dem SubType *Projekt* zugeordnet. Dadurch wird es möglich, die Zielpersonen von Eskalationen bzw. Stellvertretungen folgendermaßen zu definieren: die Rolle mit SubType *Tätigkeit* und die Gruppe vom SubType *Projekt* müssen so sein, wie bei der (normalen) Bearbeiterzuordnung dieser Aktivität (d.h. aber auch: eine dort ggf. vorkommende Rolle mit SubType *Skill-Level* ist für diese TPR nicht relevant). Eine einzige solche TPR wird dadurch für sehr viele Originalbearbeiter und für Bearbeiterzuordnung vieler Aktivitäten verwendbar. Wie in den Beispielen dargestellt, dürfen diese Bearbeiterzuordnungen sogar mittels booleschen Operationen aus mehreren Teilen zusammengesetzt sein und sich auf Prozessvariablen und Bearbeiter von Vorgängeraktivitäten beziehen (d.h. abhängige Bearbeiterzuordnungen).

Im Abschnitt 2 werden der Stand von Wissenschaft und Technik und die Herausforderungen dargestellt. Das entwickelte Konzept wird Abschnitt 3 erläutert. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick.

2 Aktuell bestehende Herausforderungen

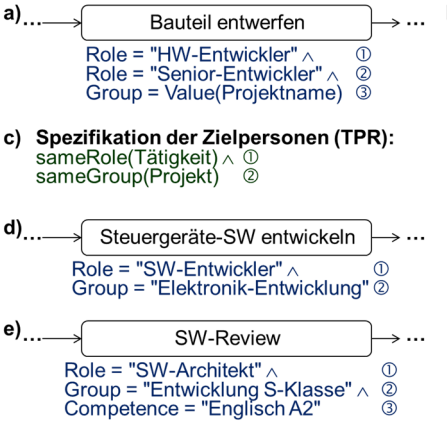
Im Folgenden werden relevante Problemstellungen anhand eines (vereinfachten, aber realitätsnahen) Anwendungsbeispiels detailliert erläutert. Dann wird der aktuelle Stand von Forschung und kommerziellen PMS dargestellt, die Anforderungen an die Festlegung der Zielpersonen beschrieben und daraus die Forschungsfrage abgeleitet.

2.1 Problemstellung und Anwendungsbeispiel

Einem Benutzer eines PMS kann mehr als eine Rolle zugeordnet sein. Deshalb kann die Definition der Zielpersonen einer Eskalation oder Stellvertretung (TPR) nicht einfach mittels Regeln wie z.B. „*selbe Rolle wie der Originalbearbeiter*“ erfolgen. Dann wäre nämlich unklar, welche seiner Rollen gemeint ist. Dasselbe gilt für Gruppen, Kompetenzen, etc. Außerdem bestehen Bearbeiterzuordnungen häufig aus mehreren Teilen, die mit booleschen Operationen verknüpft sind. Wie bereits beschrieben, sollen auch solche Bearbeiterzuordnungen Grundlage für eine TPR sein können. Hierzu müssen bestimmte Teile einer solchen Bearbeiterzuordnung referenziert werden können. Es ist also ein Mechanismus notwendig, um genau diese benötigten Teile von Bearbeiterzuordnungen in einer TPR auswählen zu können. Die daraus resultierenden Problemstellungen werden im Folgenden an Beispielen erläutert, die später bei der Darstellung des Lösungsansatzes aufgegriffen werden.

Der Mitarbeiter (User) u sei aufgrund seiner aktuellen Aufgaben Mitglied in den beiden Gruppen (Group) *Entwicklung C-Klasse* und *Entwicklung S-Klasse*. Diese Gruppenmitgliedschaften können in einer TPR nicht direkt übernommen werden, weil dann die Mitglieder beider Gruppen enthalten wären, die konkrete Eskalation oder Stellvertretung aber beispielsweise eine Aufgabe speziell in der C-Klassen-Entwicklung betrifft.

Deshalb wird nicht die Gruppenzugehörigkeit des ursprünglichen Bearbeiters verwendet, sondern die Bearbeiterzuordnung der betroffenen Aktivität. Diese ist für die Akt. *Bauteil entwerfen* in Abb. 1a dargestellt: Außer zwei Rollen (Role, siehe ①, ②) wird hier auch eine Gruppenzugehörigkeit gefordert, wobei sich der Gruppenname aus der Prozessvariablen *Projektname* dynamisch zur Runtime der Prozessinstanz ergibt ③. Abb. 1b zeigt einige der im Organisationsmodell des PMS gespeicherten Objekte einschließlich ihres SubType. Diese werden in der TPR (Abb. 1c) verwendet. Die Regel legt fest, dass die Zielpersonen dieselbe Rolle mit SubType *Tätigkeit* haben müssen ①, und derselben Gruppe mit dem SubType *Projekt* zugeordnet sein müssen ②, wie in der Bearbeiterzuordnung (Abb. 1a) festgelegt. Das Projekt hat sich dynamisch aus der Prozessvariablen *Projektname* ergeben, was in der TPR aus Abb. 1c aber nicht erkennbar ist (d.h. diese wird dadurch nicht komplizierter).



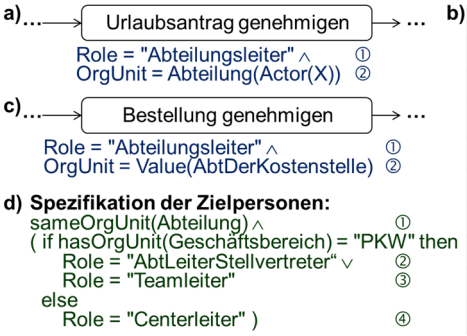
b)

Nr	Name	Type	SubType
1	HW-Entwickler	Role	Tätigkeit
2	Testfahrer	Role	Tätigkeit
3	SW-Entwickler	Role	Tätigkeit
4	SW-Architekt	Role	Tätigkeit
5	Senior-Entwickler	Role	Skill-Level
6	Entwicklung C-Klasse	Group	Projekt
7	Entwicklung E-Klasse	Group	Projekt
8	Entwicklung S-Klasse	Group	Projekt
9	Elektronik-Entwicklung	Group	Projekt
10	Betriebsrat	Group	Gremium
11	Englisch A1	Competence	Sprachkenntnis
12	Englisch A2	Competence	Sprachkenntnis

Abb. 1: a), d) und e) Bearbeiterzuordnungen, b) Objekte des Organisationsmodells, c) eine TPR

Die in Abb. 1d dargestellte Aktivität hat eine ähnliche, jedoch abweichende Bearbeiterzuordnung: So muss der Bearbeiter hier die Rolle *SW-Entwickler* haben ① und er muss kein *Senior-Entwickler* sein. Außerdem ist das zugehörige Projekt fest durch den Wert *Elektronik-Entwicklung* vorgegeben ②. Abb. 1e zeigt die Bearbeiterzuordnung der Akt. *SW-Review* eines Prozesstyps von S-Klasse-Entwicklungsprojekten. Deshalb ist außer der Rolle ① auch die Gruppe (das Projekt) ② fest vorgegeben. Zusätzlich ist noch eine bestimmte Kompetenz ③ erforderlich. Obwohl die in Abb. 1a, d und e dargestellten Bearbeiterzuordnungen recht unterschiedlich sind, kann für sie alle die in Abb. 1c dargestellte TPR verwendet werden: Diese bezieht sich auf die in den Bearbeiterzuordnungen vorkommenden Tätigkeiten (Type *Role* mit SubType *Tätigkeit*) und Projekte (Type *Group* mit SubType *Projekt*), unabhängig davon, auf welche Art und Weise diese in den Original-Bearbeiterzuordnungen festgelegt werden. Deshalb ergeben sich bei diesen Aktivitäten z.B. die unterschiedlichen Rollen *HW-Entwickler*, *SW-Entwickler* bzw. *SW-Architekt* (durch dieselbe TPR aus Abb. 1c).

In Abb. 1 wurden einige „fachliche Aktivitäten“ diskutiert. Zusätzlich zur in Abb. 1c dargestellten TPR, können für andere Aktivitäten abweichende Regeln erforderlich sein. Auch kann es unterschiedliche Regeln für Eskalationen und für Stellvertretungen geben oder mehrere Regeln für jeweils unterschiedliche Originalbearbeiter.[3] Insbesondere für „organisatorische Aktivitäten“, z.B. innerhalb von Teams oder Abteilungen (d.h. der Linienorganisation), sind typischerweise andere TPR erforderlich, weil die Zielpersonen dann z.B. Vorgesetzte sind (anstatt Projektmitglieder). So wird die in Abb. 2a dargestellte Genehmigung vom Abteilungsleiter ① desjenigen Mitarbeiters durchgeführt, der die Akt. X (Antragstellung) ausgeführt hat ②. Auch die in Abb. 2c dargestellte Bestellgenehmigung erfolgt durch einen Abteilungsleiter ①. Allerdings ergibt sich dessen Abteilung dadurch, dass aus der betroffenen Kostenstelle (wurde vom Antragsteller eingegeben) die Abteilungsbezeichnung ermittelt und in der Prozessvariablen *AbtDerKostenstelle* gespeichert wurde. Dieser Wert wird nun verwendet, um die betroffene Organisationseinheit zu ermitteln ② (z.B. Zeile 13 oder 14 in Abb. 2b). Für beide Aktivitäten gilt die in Abb. 2d dargestellte TPR: Zielpersonen müssen derselben *OrgUnit* mit SubType *Abteilung* angehören ①. Die Regel zur Festlegung ihrer Rolle hängt vom *Geschäftsbereich* des Originalbearbeiters ab: Gehört dieser zum Geschäftsbereich *PKW*, dann muss die Zielperson ein stellvertretender Abteilungsleiter ② oder ein Teamleiter ③ sein. Bei anderen Geschäftsbereichen handelt es sich um den Leiter des der Abteilung übergeordneten Centers ④. Diese Regel ist zwar recht komplex (und deshalb erfordert ihre Erstellung einen gewissen Aufwand), sie kann aber für mehrere unterschiedliche Aktivitäten verwendet werden.



b)

Nr	Name	Type	SubType
13	Sitzentwicklung	OrgUnit	Abteilung
14	Blinkerentwicklung	OrgUnit	Abteilung
15	Crash-Tests	OrgUnit	Center
16	Teamleiter	Role	Leitungsfunktion
17	Abteilungsleiter	Role	Leitungsfunktion
18	AbtLeiterStellvertreter	Role	Leitungsfunktion
19	Centerleiter	Role	Leitungsfunktion
20	PKW	OrgUnit	Geschäftsbereich
21	LKW	OrgUnit	Geschäftsbereich

Abb. 2: a) und c) Bearbeiterzuordnungen, b) Organisationsmodellobjekte und d) eine TPR

Die dargestellten Beispiele enthalten bereits abhängige Bearbeiterzuordnungen, die auf den Bearbeiter einer Vorgängeraktivität bzw. Prozessvariablen verweisen (Abb. 2a und c). Das zu entwickelnde Konzept muss zusätzlich weitere Arten abhängiger Bearbeiterzuordnungen unterstützen. Die Schwierigkeit hierbei ist, dass eine abhängige Bearbeiterzuordnung wie z.B. „*selber Bearbeiter wie Akt. X*“ keine Rollen, Gruppen, etc. spezifiziert, die durch eine TPR referenziert werden könnten. Eine weitere bereits dargestellte Anforderung ist, dass in den TPR Fallunterscheidungen vorkommen können (if/else in Abb. 2d). Schließlich sind TPR, ebenso wie Bearbeiterzuordnungen, oft mittels boole-

[3] Auf die Darstellung solcher Fälle wurde hier zur Verbesserung der Übersichtlichkeit verzichtet.

scher Operationen aus mehreren Teilen zusammengesetzt. Durch die Einführung der benutzerdefinierten SubTypes werden einzelne Teile solcher Bearbeiterzuordnungen referenzierbar. Dadurch entfällt der Aufwand für GP-Designer, für jede Kombination von vielen unterschiedlichen Aktivitäten und zahlreichen unterschiedlichen Personengruppen (z.B. Geschäftsbereiche, Mitarbeiterarten), jeweils eine separate TPR zu erstellen.

2.2 Stand von Forschung und Technik

Im Folgenden wird wissenschaftliche Literatur zu Eskalationen und Stellvertretungen, sowie der aktuelle Stand einiger kommerzieller PMS darstellt. Dabei wurden Bizagi Studio V.11.2.3 [Bi21], IBM Business Automation Workflow V20.0.0.2 [IBM20], K2 Cloud V.4.0 [K21] und Signavio Workflow Accelerator V.13.6.0 [Si21] betrachtet.

Eskalationen

Eskalationen werden in der wissenschaftlichen Literatur zu PMS zwar erwähnt (z.B. in [Gr08, Ru05]), es gibt aber keine Arbeiten, die sich hiermit umfassend beschäftigen. Deshalb werden im Folgenden Arbeiten vorgestellt, die spezielle Arten bzw. Aspekte von Eskalationen betrachten: [ARD07] verwendet erwartete Zeiten für die Beendigung von Aktivitäten, um schon vor dem Verpassen einer Frist, geeignet reagieren zu können. Hierzu wird ein sehr mächtiger Eskalationsmechanismus, speziell für das betrachtete Szenario der Überlastung eines Call-Centers, entwickelt. [PR98] minimiert die Anzahl an Eskalationen, indem Deadlines automatisch angepasst werden, um Verzögerungen auszugleichen. Die von uns betrachtete Fragestellung (d.h. die effiziente Modellierung der Zielpersonen einer Eskalation) wird in all diesen Arbeiten nicht erwähnt.

Die betrachteten kommerziellen PMS bieten unterschiedliche Funktionalitäten. So ermöglicht Bizagi keine Eskalationen. Bei K2 kann ein einzelner Benutzer als Zielperson festgelegt werden, der abhängig von der betroffenen Aktivität ist. Bei Signavio sind auch mehrere Zielpersonen möglich. Auch IBM ermöglicht mehrere Zielpersonen, und es können zudem mehrere Eskalationen (parallel und nacheinander) definiert werden.

Stellvertretungen

[Ba09] definiert Anforderungen, wie z.B. kontext-abhängige und mehrstufige Stellvertretungen, oder eine Aktivität dem Stellvertreter bei Rückkehr des Originalbearbeiters wieder zu entziehen. Hierfür werden Algorithmen zur Berechnung der Stellvertreter vorgestellt. Die Arbeit enthält jedoch kein Konzept zur Modellierung solcher Zielpersonen. Auch andere Arbeiten zum Thema Stellvertretungen machen hierzu keine Vorschläge: In [Mu04] werden Stellvertreter mittels Rollen definiert. [Ru05] betrachtet Stellvertretungen als vom PMS durchgeführte Delegation. [HD05] erwähnt die Notwendigkeit von Stellvertretungen, ohne Lösungskonzepte vorzustellen. [RM98] fordert „kontextbezogene Stellvertretungen“, betrachtet jedoch die Modellierung solcher Stellvertreterregelungen nicht. [HS99] definiert Ersatzressourcen (d.h. auch Bearbeiter) mittels einer SQL-artigen Sprache. [DW15] ermittelt die optimale Ersatzressource mittels Process-Mining.

Bizagi unterstützt Stellvertretungen nicht. K2 ermöglicht unterschiedliche Stellvertreter für verschiedene Aktivitäten. Bei IBM kann für jeden Benutzer eine Liste von Stellvertretern definiert werden. Die Stellvertreter können aber nicht abhängig von der betroffenen Aktivität oder von Prozessdaten sein. Zudem wird nur der erste anwesende Stellvertreter dieser Liste berücksichtigt. Wie auch bei den Eskalationsmechanismen bietet keines der PMS effiziente Regeln zur Festlegung der Zielpersonen, die zudem sehr leistungsfähig sind (d.h. komplexe, vom Prozesskontext abhängige Regeln).

2.3 Anforderungen an die Festlegung von Zielpersonen und Forschungsfrage

Nachfolgend werden die im Abschnitt 2.1 erläuterten Anforderungen (Requirements) zusammengefasst und anschließend die Forschungsfrage vorgestellt.

- **Req1:** Es soll für den GP-Designer möglich sein, die Zielpersonen mit geringem Aufwand (d.h. effizient) zu modellieren. Deshalb muss eine einzelne solche Regel (TPR) für mehrere Aktivitäten und Originalbearbeiter verwendbar sein.
- **Req2:** Es sollen beliebig viele unterschiedliche Regeln definierbar sein, die jeweils für einen unterschiedlichen Kontext gültig sind. Dieser muss beliebig definierbar sein, d.h. sich z.B. auf die betroffenen Originalbearbeiter und Aktivitäten, sowie den Prozesskontext (die aktuellen Anwendungsdaten der Prozessinstanz) beziehen können.
- **Req3:** Die Bearbeiterzuordnung einer Aktivität darf (weiterhin) beliebig komplex sein, d.h. sie kann mittels boolescher Operationen aus mehreren Teilen zusammengesetzt sein, und Fallunterscheidungen enthalten.
- **Req4:** Teile einer Bearbeiterzuordnung können abhängig z.B. (vom Bearbeiter) von Vorgängeraktivitäten oder von Prozessdaten sein.
- **Req5:** Die Definition der Zielpersonen (TPR) kann sich auf die Bearbeiterzuordnung der Aktivität beziehen und muss dort beliebige organisatorische Objekte (z.B. eine Rolle) referenzieren können. Dies muss auch möglich sein, wenn die Bearbeiterzuordnung komplex ist (Req3) oder abhängige Teile enthält (Req4).

Diese Anforderungen stellen ein komplexes Problem dar, das in der wissenschaftlichen Literatur und in kommerziellen PMS (vgl. Abschnitt 2.2) noch ungelöst ist. Deshalb beschäftigt sich die vorliegende Arbeit mit folgender Forschungsfrage: Wie kann die Festlegung der Zielpersonen von Eskalationen und Stellvertretungen so gestaltet werden, dass eine große Mächtigkeit erreicht wird, für Aktivitäten weiterhin die benötigten Bearbeiterzuordnungen verwendbar sind, und dennoch ein möglichst geringer Aufwand bei der Erstellung entsprechender Regeln entsteht?

3 Konzept zur effizienten Modellierung der Zielpersonen

Im Folgenden wird erläutert, wie Organisationsmodell-Objekte inkl. SubType definiert und später in einer TPR verwendet werden. Außerdem wird auf die Aspekte komplexer (d.h. zusammengesetzter) TPR und abhängiger Bearbeiterzuordnungen eingegangen.

3.1 Erweiterung und Verwendung des Organisationsmodells

Wie bereits erwähnt, basiert der hier vorgestellte Ansatz darauf, dass die Objekte des Organisationsmodells einem benutzerdefinierten SubType zugeordnet werden. Es ist also die Festlegung zusätzlicher Information erforderlich. Diese Information wird bewusst nicht als Teil von Bearbeiterzuordnungen definiert, weil dies sonst bei jeder Bearbeiterzuordnungen jeder einzelnen Aktivität erfolgen müsste, was zu einem großen Aufwand für die GP-Designer führen würde. Das bedeutet, die Bearbeiterzuordnungen bleiben gegenüber einer „klassischen GP-Modellierung" unverändert. Die Festlegung der Zusatzinformation SubType erfolgt im Organisationsmodell bei der Erstellung der Organisationsmodell-Objekte (z.B. einer Rolle). Solche Objekte werden selten neu erstellt und sie werden selten verändert (ihnen werden lediglich öfters neue oder andere Benutzer zugeordnet). Deshalb erfordert die Festlegung der SubTypes nur einen geringen Aufwand.

Das diesem Beitrag zugrundeliegende Organisationsmodell unterscheidet die Objekttypen Role, Group, Competence und OrgUnit (vgl. [Ru05]). Diese Objekttypen werden jedoch weitgehend auf identische Art und Weise verwendet, so dass der Ansatz leicht um beliebige weitere Typen erweitert werden kann. Jedes Objekt des Organisationsmodells kann einem SubType zugeordnet werden (z.B. die Rolle *HW-Entwickler* dem SubType *Tätigkeit*, vgl. Abb. 1b). Bei der Spezifikation einer TPR kann darauf mittels vom PMS bereitgestellten Funktionen Bezug genommen werden. So legt *sameRole(S)* fest (vgl. Abb. 1c), dass die Zielpersonen diejenige Rolle des SubType *S* besitzen müssen, die in der Bearbeiterzuordnung dieser Aktivität festgelegt wurde (z.B. bei SubType = *Tätigkeit* die Rolle *HW-Entwickler*, vgl. Abb. 1a). Ähnlich funktioniert die Funktion *hasRole(R)*, mit der abgefragt werden kann, ob der Originalbearbeiter der Aktivität der Rolle *R* zugeordnet ist oder nicht. Dies wird z.B. für Fallunterscheidungen (if) genutzt. Dem GP-Designer stehen analoge Funktionen für Organisationsmodell-Objekte des Typs Group (d.h. *sameGroup(S)* und *hasGroup(G)*), Competence und OrgUnit zur Verfügung.

Bei der Auswertung einer TPR (folgt in Abschnitt 3.2) werden intern nicht die oben dargestellten Funktionen verwendet. Stattdessen werden die TPR nach Abschluss der Modellierung (d.h. zur Buildtime) automatisch in generic Target Person Rules (*gTPR*) umgewandelt, die auf Funktionen basieren, in deren Namen der Typ des jeweils betroffenen Organisationsobjekts nicht enthalten ist. Diese Funktionen werden bei Anfragen an die Organisationsmodell-Datenbank zur Ermittlung der entsprechenden Benutzer verwendet. Eine dieser generischen Funktionen ist *sameOrgObject(Type, SubType)*. Diese wird z.B. verwendet, um den vom GP-Designer modellierten Ausdruck *sameRole(Tätigkeit)* in *sameOrgObject(Role, Tätigkeit)* umzuwandeln. Die andere generische Funktion ist *hasOrgObject(Type, SubType)*. Sie ist erforderlich, um z.B. *hasGroup(Betriebsrat)* in *hasOrgObject(Group, Betriebsrat)* umzuwandeln. Vorteil dieser generischen Funktionen ist, dass die Schnittstelle der Organisationsdatenbank nur diese beiden Funktionen unterstützen muss, und dennoch beliebige Typen von Organisationsmodell-Objekten verwendet werden können. Die im Weiteren dargestellten Inhalte bleiben ebenfalls für beliebige zusätzliche Typen unverändert. Die Umformung der vom GP-Designer modellierten TPR in die gTPR mit generischen Funktionen kann

automatisch erfolgen, weil beide Arten von Ausdrücken inhaltlich dasselbe beschreiben. Dem GP-Designer sollten aber (weiterhin) „typisierte" Funktionen (z.B. *sameRole(R)*) zur Verfügung gestellt werden, weil diese Personen solche Funktionen gewöhnt sind.

Zwischen Mitarbeitern bzw. den von ihnen besetzten Stellen existiert eine Hierarchie. Diese wird von normalen Bearbeiterzuordnungen verwendet (z.B. „*Vorgesetzter vom Bearbeiter der Akt. X*"). Solche Hierarchien sind, abhängig vom konkreten Organisationsmetamodell des verwendeten PMS, evtl. bereits vorhanden. Sie können im hier vorgestellten Ansatz zur Ermittlung von Zielpersonen von Eskalationen und Stellvertretungen in einer TPR ebenfalls verwendet werden.

Es kann der Eindruck entstehen, dass die vorgeschlagene Lösung unnötig komplex ist, weil ein Objekt des Organisationsmodells nun zwei „Kategorien" zugeordnet ist (z.B. dem Type *Role* und SubType *Tätigkeit*). Die Alternative hierzu wäre, auf die Einführung der SubTypes zu verzichten und stattdessen zu ermöglichen, dass der Benutzer zusätzliche Types selbst definiert. Dies ist allerdings kaum umsetzbar, weil die Metamodelle von PMS üblicherweise von einer fest vorgegebenen Menge an Typen von Organisationsmodell-Objekten ausgehen (vgl. [BL20]). Auch die wissenschaftliche Literatur betrachtet üblicherweise keine frei durch den Benutzer definierbaren Typen von Organisationsobjekten [Ru05]. Dies wäre auch aufwendig und kaum nützlich, weil in einer Bearbeiterzuordnung dann anstatt „*Role = HW-Entwickler*" der selbst definierte Type verwendet werden müsste, also z.B. „*Role_Tätigkeit = HW-Entwickler*". Da der SubType bei „normalen" Bearbeiterzuordnung jedoch gar nicht relevant ist, würde dies deren Definition sogar verkomplizieren. Der Vorteil bei der Festlegung einer TPR fällt im Vergleich hierzu kaum ins Gewicht, weil vergleichsweise wenige TPR definiert werden müssen. Dort ergäbe sich in diesem Beispiel zudem ein nicht wirklich einfacherer Ausdruck, nämlich *same(Role_Tätigkeit)* anstatt *sameRole(Tätigkeit)*. Generell ist es bei beiden Alternativen möglich, dass der GP-Designer sowohl Funktionen wie *sameRole*, als auch die benötigten Werte von Type und SubType graphisch in einer GUI aus einer Liste auswählt. Für den Benutzer ist der hier vorgestellte Ansatz also einfach handhabbar

3.2 Spezifikation und Ermittlung der Zielpersonen

Im Folgenden werden die Regeln zur Festlegung von Zielpersonen vollständig eingeführt und es wird beschrieben, wie diese PMS-intern verwendet werden.

Beim GP-Design wird eine Menge *ESR* (Escalation and Substitution Rules) zur Definition von Zielpersonen festgelegt: ESR = $\{R_1, R_2, \ldots\}$. Jede dieser Regeln besteht aus einer Bedingung Cond (Condition), die ihren Gültigkeitsbereich definiert, und einer Regel TPR zur Festlegung der Zielpersonen: R_i = (Cond, TPR). Hierbei ist die Bedingung *Cond* nicht weiter problematisch. Dies ist eine boolesche Regel, die beliebige Attribute verwenden kann (z.B. *Cond = „ActivityClass = Entwicklungstask ∧ hasOrgUnit(Geschäftsbereich) = PKW"*). Diese können auch benutzerdefiniert sein, wie z.B. *ActivityClass* (ein dieser Aktivität zugeordneter Klassenname). Außerdem können

z.B. organisatorische Objekte oder Werte von Prozessvariablen verwendet werden, und es kann definiert werden, ob die Regel für Eskalationen, Stellvertretungen oder für beides gültig ist. Ergibt die Bedingung Cond den Wert True, ist die TPR im aktuell vorliegenden Fall anzuwenden. Im Folgenden liegt der Fokus auf dieser Regel TPR.

Zur Buildtime definiert der GP-Designer Bearbeiterzuordnungen *AA* (Actor Assignment) für Aktivitäten a, z.B. *AA(a) = „Role = HW-Entwickler“*. In diesem Fall hat die Rolle *HW-Entwickler* den SubType *Tätigkeit* (vgl. Abb. 1b). Außerdem definiert er *TPR*, z.B. für diese Aktivität a die Regel *TPR = „sameRole(Tätigkeit)“*. Selbstverständlich sind hier auch komplexere Regeln möglich (vgl. Abschnitt 3.3 und 3.4). Wie bereits erläutert, werden die modellierten TPR in ein nicht-typisiertes (d.h. generisches) Format umgewandelt, in diesem Beispiel *gTPR* = „*sameOrgObject(Role, Tätigkeit)“*.

Ist nun zur Runtime eine Eskalation oder Stellvertretung für den Originalbearbeiter *OriginalActor* dieser Aktivität a erforderlich, so ermittelt das PMS anhand deren Bedingung Cond die passende Regel R_i. Die daraus resultierende gTPR wird verwendet, um diejenigen Organisationsmodell-Objekte zu ermitteln, denen die Zielpersonen angehören müssen. Im obigen Beispiel ergibt sich also aus *gTPR = „sameOrgObject(Role, Tätigkeit)“* die Rolle *HW-Entwickler*, weil dies diejenige Rolle mit SubType *Tätigkeit* ist, die in *AA(a)* enthalten ist. Dieses konkrete Organisationsmodell-Objekt *O = HW-Entwickler* wird verwendet, um aus der gTPR eine rTPR (resolved TPR) abzuleiten. Diese hat die Form *OrgObjectType = O*, im dargestellten Beispiel also *rTPR = „Role=HW-Entwickler“*. Diese rTPR wird verwendet, um die entsprechenden Zielpersonen mit der Funktion *getUsers(rTPR)* von der Organisationsmodell-Datenbank zu erfragen, hier also mit *getUsers("Role=HW-Entwickler")*. Funktionen wie *hasRole()*, *hasGroup()* werden analog behandelt, wobei diese aber True oder False zurückliefern, abhängig davon, ob der Originalbearbeiter *OriginalActor* dem ermittelten Organisationsobjekt *O* angehört (z.B. der Rolle *HW-Entwickler*).

Selbstverständlich können bei der Festlegung einer TPR auch „klassische“ Regeln verwendet werden (d.h. so wie bei normalen Bearbeiterzuordnungen ohne SubType). Dann entfällt die eben beschriebene Umformung. Beispielsweise ergeben sich für die TPR *„Role = HW-Entwickler“* die identischen gTPR und rTPR und damit direkt die Anfrage *getUsers("Role=HW-Entwickler")*.

3.3 Komplexe Regeln für Bearbeiterzuordnungen

Die Anforderung Req3 besagt, dass eine TPR und damit eine gTPR eine komplexe (d.h. zusammengesetzte) Regel sein kann. Dies wird folgendermaßen behandelt: Wenn eine gTPR eine Bedingung (if) (vgl. Abb. 2d) enthält, so wird diese zuerst aufgelöst, indem der für den betroffenen Bearbeiter *OriginalActor* relevante Fall ausgewählt wird (z.B. anhand der aktuellen Werte von Prozessvariablen oder vorheriger Bearbeiter). Dadurch entsteht eine Formel, die ausschließlich boolesche Operationen enthält. Beispielsweise entsteht aus der in Abb. 2d dargestellten TPR im else-Fall *gTPR = „sameOrg-*

Object(OrgUnit, Abteilung) ∧ Role = Centerleiter". Die mit booleschen Operationen verknüpften Teile werden nun, so wie bereits erläutert, einzeln umgeformt, um die rTPR zu berechnen. Falls sich z.B. bei der Bearbeiterzuordnung aus Abb. 2a aus dem Bearbeiter dieser Aktivität die Abteilung *Sitzentwicklung* ergibt, entsteht die *rTPR = „OrgUnit = Sitzentwicklung ∧ Role = Centerleiter"*, wobei der zweite Teil unverändert aus der *gTPR* übernommen wurde. Diese rTPR wird dann von der Funktion *getUsers(rTPR)* zur Abfrage der gesuchten Benutzer verwendet.

Es wäre einfach, zu fordern, dass für alle Regelteile einer TPR, die sich auf eine Bearbeiterzuordnung beziehen (z.B. mittels *sameRole()*), eine entsprechende Teilbedingung auch tatsächlich in jeder referenzierten Bearbeiterzuordnung enthalten sein muss. Dann wäre diese TPR jedoch nicht für Aktivitäten verwendbar, bei denen einer dieser Teile fehlt. Die Folge wäre, dass hierfür eine weitere TPR nötig wird, also mehr Regeln und mehr Aufwand für die GP-Designer entstehen. Das folgende Beispiel zeigt, dass solche zusätzlichen Regeln nicht unbedingt erforderlich sind: Angenommen, für eine Aktivität sind keine besonderen Sprachkenntnisse (wie z.B. *Englisch A2*, vgl. Abb. 1b) erforderlich. Außerdem existiert bereits eine (ansonsten passende) TPR, welche dieselben Sprachkenntnisse (*Competence* mit SubType *Sprachkenntnis*) fordert, wie bei der Bearbeiterzuordnung. Da in dieser jedoch gar keine Sprachkenntnisse gefordert sind, sind diese auch für etwaige Stellvertreter nicht notwendig. Die bereits vorhandene TPR kann also auch für solche Aktivitäten verwendet werden, indem diese Bedingung geeignet ignoriert wird. Im Folgenden wird erläutert, wie ein PMS das umsetzen kann.

Da jede boolesche Formel in die Disjunktive Normalform (DNF) umgeformt werden kann (automatisch vom PMS), wird im Folgenden angenommen, dass die gTPR in diesem Format vorliegt, z.B.: *gTPR = (A1 ∧ A2 ∧ A3) ∨ (B1 ∧ B2) ∨ (C1)*

Ein in der *gTPR* enthaltener Teilausdruck (z.B. *sameOrgObject(Type, SubType)*) kann sich auf einen Objekttyp *Type* (z.B. eine Rolle) mit einem SubType (z.B. *Tätigkeit*) beziehen, der in der Bearbeiterzuordnung dieser Aktivität nicht enthalten ist. Es wird also etwas „nicht Vorhandenes" referenziert. Dies wird vom PMS folgendermaßen behandelt:

1. Trifft dies für einen Teil einer Konjunktion (d.h. einer AND-Verknüpfung) zu, aber nicht für alle ihre Teilausdrücke (z.B. *A2* der in DNF dargestellten Regel), so wird dieser Teilausdruck ignoriert (d.h. aus der Regel gestrichen). Damit ist nur noch der Rest der Konjunktion relevant, im Beispiel also *A1 ∧ A3*, womit sich die Gesamtregel *gTPR' = (A1 ∧ A3) ∨ (B1 ∧ B2) ∨ (C1)* ergibt. Dies ist der Fall, der im obigen Beispiel der nicht relevanten Sprachkenntnisse greift.
2. Trifft dies für alle Teile einer Konjunktion zu (z.B. für *B1* und *B2* oder für *C1*, dann werden diese Teile ignoriert, d.h. die gesamte Konjunktion entfällt, weil eine nicht verwendbare Bedingung nicht zu zusätzlichen Zielpersonen führen soll. Bei nicht relevantem *B1* und *B2* verbleibt von obiger Formel beispielsweise *gTPR' = (A1 ∧ A2 ∧ A3) ∨ (C1)*.
3. Trifft dies für alle Teile einer Regel zu, so ist sie für diese Aktivität nicht verwendbar. Es liegt also ein Modellierungsfehler vor. Dieser kann bereits zur Buildtime automatisch erkannt und damit verhindert werden.

3.4 Berücksichtigung abhängiger Bearbeiterzuordnungen

Entsprechend der Anforderung Req4 sollen auch abhängige Bearbeiterzuordnungen durch eine TPR referenziert werden können. So soll die Akt. Y in Abb. 3a vom selben Bearbeiter ausgeführt werden, wie die Vorgängerakt. X. Allerdings enthält der Ausdruck (Actor Assignment) *AA(Y) = sameActor(X)* keinen Rollennamen, der z.B. von dem in der TPR enthaltenen Teil *sameRole(Tätigkeit)* referenziert werden könnte (vgl. Abb. 3b). Der tatsächliche Bearbeiter der Akt. X und damit auch der Akt. Y steht jedoch vor dem Start von Y bereits fest. Es ist aber nicht möglich, einfach die ihm zugeordnete Rolle zu verwenden, da er mehrere Rollen mit SubTyp *Tätigkeit* innehaben kann (z.B. *SW-Entwickler* und *SW-Architekt*). Dasselbe gilt für andere Typen organisatorischer Objekte, wie Kompetenzen (z.B. *Englisch A1* und *Französisch B2*).

Abb. 3: a) Prozess mit abhängigen Bearbeiterzuordnungen und b) eine Target Person Rule (TPR)

Um unabhängige Bearbeiterzuordnungen zu erhalten, wird aus der (ggf. abhängigen) Bearbeiterzuordnung AA(a) eine unabhängige (independent) Bearbeiterzuordnung *iAA(a)* abgeleitet, sobald die Akt. a ausführbar wird. Diese neue Regel iAA(a) wird dann von Folgeaktivitäten verwendet, welche die Bearbeiterzuordnung der Akt. a referenzieren. i) Im einfachsten Fall enthält AA(a) keine Abhängigkeiten und wird unmittelbar übernommen, d.h. iAA(a) = AA(a). ii) Enthält eine Bearbeiterzuordnung Prozessvariablen, so werden diese durch ihren aktuellen Wert ersetzt. Der in Abb. 3a dargestellte Teil von AA(X) *Group = Value(FzgProjekt)* wird also umgeformt in z.B. *Group = Entwicklung C-Klasse*. iii) Im Falle einer Bearbeiterzuordnungen des Typs *AA(Y) = sameActor(X)* wird iAA(Y) gebildet, indem die unabhängige Bearbeiterzuordnung (*iAA*) der referenzierten Aktivität X übernommen wird: iAA(Y) = iAA(X). Im Beispiel aus Abb. 3a ergibt sich also z.B. *iAA(Y) = „Role = SW-Entwickler ∧ Group = Entwicklung C-Klasse“*. Die hierdurch entstandene unabhängige Bearbeiterzuordnung iAA(Y) kann dann von der in Abb. 3b dargestellten TPR, so wie in Abschnitt 3.3 beschrieben, problemlos verwendet werden. Diese Vorgehensweise kann über beliebig viele Stufen hinweg angewandt werden. So ergibt sich im Beispiel aus Abb. 3a iAA(Z) analog als iAA(Y).

Organisatorische Objekte des Typs *OrgUnit* stellen eine Besonderheit dar, weil jeder Benutzer üblicherweise maximal einer *OrgUnit* jeder Ebene angehören kann, d.h. maximal einem Team, einer Abteilung, einem Center, etc. Damit entfällt das bereits erläuterte Problem, dass (für den vorgegebenen SubType) z.B. die Rolle des tatsächlichen Bearbeiters der referenzierten Akt. X nicht eindeutig ist. Deshalb ist für den Fall *OrgUnit* ein verbessertes Verhalten realisierbar: Beim Start der Akt. Y aus Abb. 3a ist der Bearbeiter (User) u der Akt. X bereits bekannt. Dieser ist auch der regulär vorgesehene Bearbeiter für die Akt. Y. Der in der TPR der Akt. Y enthaltene Teil *sameOrgUnit(Abteilung)* (Abb. 3b) kann sich also konkret auf diesen Bearbeiter u beziehen. Gehört er z.B. zur Abteilung *Sitzentwicklung*, so muss auch die Zielperson dieser Abteilung angehören. Die

rTRP erhält deshalb die Teilbedingung *OrgUnit = Sitzentwicklung*. Dies ist möglich, obwohl weder die reguläre Bearbeiterzuordnung AA(Y) noch iAA(Y) eine Vorgabe für die *OrgUnit* enthält. So wird eine genauere Anwendung der TPR möglich, als wenn dieser Teil durch die in Abschnitt 3.3 vorgestellte „Standard-Vorgehensweise" ignoriert wird. Ob dies überhaupt gewünscht ist, sollte der GP-Designer jedoch auswählen können.

Eine ähnliche Vorgehensweise ist möglich, wenn die Akt. Y einem Vorgesetzten des Bearbeiters u der Akt. X zugeordnet ist. Dies ist ein relevanter Fall, weil solche Bearbeiterzuordnungen bei Prüf- oder Genehmigungsaufgaben häufig vorkommen. Der Unterschied zum eben vorgestellten Fall ist, dass mittels des Organisationsmodells, ausgehend vom Bearbeiter u, der Vorgesetzte u' der geforderten Ebene ermittelt wird (z.B. der Teamleiter). Dessen Abteilung wird dann von der rTPR der Akt. Y referenziert. Analog kann auch bei anderen Bearbeiterzuordnungen vorgegangen werden, die einen von u abhängigen, aber eindeutigen Benutzer ergeben, wie z.B. „*Frauenbeauftragter von …*".

Eine Bearbeiterzuordnung kann sich auch auf die Organisationseinheit einer Vorgängerakt. X beziehen. So kann AA(Y) einen Teil *sameOrgUnit(X, Abteilung)* enthalten. Dieser spezifiziert diejenigen Personen, die derselben *Abteilung* (dem SubType) wie der Bearbeiter der Akt. X angehören. Auch dies ist ein relevanter Fall, insb. im Kombination mit dem 4-Augen-Prinzip, wenn z.B. eine Überprüfung durch einen anderen Mitarbeiter derselben Abteilung erfolgen soll. In solchen Fällen wird iAA(Y) erzeugt, indem die *OrgUnit* (dieser Ebene) des tatsächlichen Bearbeiters der Akt. X aus dem Organisationsmodell ermittelt wird (z.B. die Abteilung *Blinkerentwicklung*). Diese ersetzt dann den entsprechenden Teil aus der Regel AA(Y), so dass sich z.B. *iAA(Y) = „Abteilung = Blinkerentwicklung ∧ nicht User43"* ergibt (*User43* hat die Akt. X bearbeitet). Die nun in iAA(Y) spezifizierte Abteilung Blinkerentwicklung kann durch die in Abb. 3b dargestellte TPR problemlos referenziert werden (mittels *sameOrgUnit(Abteilung)*).

4 Zusammenfassung und Ausblick

Eskalationen und Stellvertretungen sind in der Praxis wichtig, um geeignet auf die verzögerte Bearbeitung von Aktivitäten und Abwesenheit regulärer Bearbeitern zu reagieren. Deshalb wird dies von einigen kommerzielle PMS unterstützt. Allerdings werden meist nur sehr einfache Regeln (entspricht einer TPR) zur Festlegung der entsprechenden Zielpersonen angeboten. Da zudem keine ausreichend mächtigen und dennoch leicht handhabbaren wissenschaftlichen Ansätze existieren, wird in diesem Beitrag ein entsprechendes Konzept vorgestellt. Es basiert auf der Einführung eines SubType für organisatorische Objekte. Dieser ermöglicht die Referenzierung bestimmter Teile einer regulären Bearbeiterzuordnung durch eine TPR auf eine Art und Weise, welche die Verwendung der TPR für viele unterschiedliche Aktivitäten erlaubt. Dies ist sogar bei komplexen (d.h. zusammengesetzten) und abhängigen Bearbeiterzuordnungen bzw. TPR möglich.

Da die Ermittlung der Zielpersonen aus einer TPR algorithmisch recht einfach ist, wird eine prototypische Umsetzung kaum zu neuen Erkenntnissen führen. Interessanter wäre

eine Evaluation, wie leicht die GP-Designer die TPR definieren können, und ob mit dem Ansatz alle Anforderungen aus der Praxis erfüllt werden können.

Literaturverzeichnis

[ARD07] Aalst, W. M. van der; Rosemann, M.; Dumas, M.: Deadline-based Escalation in Process-Aware Information Systems. In Decision Support Systems, 2007; S. 492–511.

[Ba09] Bauer, T.: Stellvertreterregelungen für Task-Bearbeiter in prozessorientierten Applikationen. In Datenbank-Spektrum, 2009, 9; S. 40–51.

[Bi21] Bizagi 11.2.3 BPM Suite User Guide. https://help.bizagi.com/bpm-suite/en/11.2.3/, 10.2.2021.

[BL20] Bauer, T.; Laue, R.: Stand der anwendungsnahen Forschung und Technik für die organisatorische Perspektive von Geschäftsprozessen. In Proc. Informatik 2020, 6. Workshop zum Stand, den Herausforderungen und Impulsen des Geschäftsprozess-managements, 2020; 605-619.

[DW15] Dulai, T.; Werner-Stark, A.: A Database-Oriented Workflow Scheduler with Historical Data and Resource Substitution Possibilities. In Proc. 4th Int. Conf. on Operations Research and Enterprise Systems, 2015; S. 325–330.

[Gr08] Großkopf, A.: An Extended Resource Information Layer for BPMN, Hasso-Plattner-Institute for IT Systems Engineering, Potsdam, 2008.

[HD05] Hochmüller, E.; Dobrovnik, M.: Flexibility Issues in Workflow Management Systems. In Proc. Business Process Modeling, Development and Support, 2005, 5.

[HS99] Huang, Y. N.; Shan, M. C.: Policies in a Resource Manager of Workflow Systems: Modeling, Enforcement and Management. In Proc. 15th Int. Conf. on Data Engineering, 1999.

[IBM20] IBM Business Automation Workflow V20.0.0.2 Documentation. https://www.ibm.com/support/knowledgecenter/en/SS8JB4_20.x, 10.2.2021.

[K21] K2 Cloud: Low Code Digital Process Automation. https://www.nintex.com/process-automation/k2-software/, 10.2.2021.

[Mu04] Mühlen, M. Zur: Organizational Management in Workflow Applications – Issues and Perspectives. In Information Technology and Management Journal, 2004, 5.

[PR98] Panagos, E.; Rabinovich, M.: Reducing Escalation-Related Costs in WFMSs. In Workflow Management Systems and Interoperability, 1998; S. 107–128.

[RM98] Rosemann, M.; Mühlen, M. Zur: Modellierung der Aufbauorganisation in Workflow-Management-Systemen. In EMISA-Forum, 1998; S. 78–86.

[Ru05] Russell, N. et al.: Workflow Resource Patterns: Identification, Representation and Tool Support. In Proc. Int. Conf. on Advanced Information Systems Engineering, 2005.

[Si21] Signavio Workflow Accelerator. https://www.signavio.com/products/workflow-accelerator, 10.2.2021.