

HCI-Patterns für kollaborative Unternehmensmodellierung am Multi-Touch-Tisch

Anne Gutschmidt¹, Valentina Sauer², Kurt Sandkuhl³

Abstract: Sollen Unternehmensmodelle kollaborativ durch die Stakeholder selbst erstellt werden, bieten sich insbesondere Multi-Touch-Tische (MTT) als nützliches digitales Werkzeug an. Für die Gestaltung von Software und insbesondere deren Benutzungsoberfläche sollte auf bewährte Lösungen - sogenannte HCI-Patterns - zurückgegriffen werden. Das von uns betrachtete Anwendungsfeld bringt jedoch besondere Anforderungen an die Modellierungssoftware für einen Multi-Touch-Tisch mit sich. In einer Studie, angelehnt an die Repertory-Grid-Technik, ließen wir acht Modellierungsexperten, die wir in drei Gruppen einteilten, vier verschiedene Modellierungsprogramme vergleichen und ermittelten so wesentliche Merkmale von Modellierungssoftware für MTT. Diese haben wir mit Katalogen von HCI-Patterns abgeglichen und die Patterns selektiert, die am besten auf die von den Experten geäußerten Bedürfnisse passen.

Keywords: Unternehmensmodellierung; kollaboratives Modellieren; Multi-Touch-Tisch; HCI-Patterns; Modellierungssoftware; Studie

1 Einleitung

Unternehmensmodelle dienen Organisationen dazu, Wissen wie z.B. eigene Strukturen, Abläufe oder Verantwortlichkeiten festzuhalten. Mit ihrer Hilfe kann sowohl der Ist- als auch der Soll-Zustand eines Unternehmens dargestellt werden [Sa14]. Angesichts sich ständig verändernder Rahmenbedingungen, insbesondere durch die Digitalisierung von Arbeit und Produkten, bietet die Unternehmensmodellierung folglich eine große Hilfestellung. Die Partizipative Unternehmensmodellierung (PUM) sieht vor, dass die Mitglieder einer Organisation die Modelle selbst erstellen. Dabei arbeiten sie gemeinsam in kleinen Teams. Die PUM gilt insbesondere dann als sinnvoll, wenn sich verschiedene Stakeholdergruppen auf eine gemeinsame Sichtweise einigen müssen. Man erhofft sich durch das kollaborative Modellieren der tatsächlich Betroffenen eine höhere Qualität und auch Akzeptanz der Modelle [SP18]. Als Werkzeuge zur Modellierung kommen viele in Frage, wie z.B. Whiteboards, wobei digitale Werkzeuge zunehmend als attraktiv empfunden werden [Sa14], da sie einfaches Speichern, Editieren und Teilen der Modelle ermöglichen [MH14, PH09].

¹ Universität Rostock, Institut für Informatik, Lehrstuhl Softwaretechnik, Albert-Einstein-Straße 22, 18059 Rostock, Deutschland anne.gutschmidt@uni-rostock.de

² Universität Rostock, Institut für Informatik, Albert-Einstein-Straße 22, 18059 Rostock, Deutschland valentina.sauer@uni-rostock.de

³ Universität Rostock, Institut für Informatik, Lehrstuhl Wirtschaftsinformatik, Albert-Einstein-Straße 22, 18059 Rostock, Deutschland kurt.sandkuhl@uni-rostock.de

Insbesondere mit Multi-Touch-Tischen (MTT) kann gleichzeitiges Arbeiten mehrerer Nutzer auf einer geteilten Arbeitsfläche realisiert werden. In einer früheren Studie wurde bereits gezeigt, dass der MTT für kleinere Modellierungsaufgaben ebenso gut geeignet ist wie ein Whiteboard [Gu18b, Gu18a]. Das Erstellen von Unternehmensmodellen, aber auch die besonders heterogene Gruppe von Nutzern, die wenige bis keine Modellierungsvorkenntnisse besitzt, bringen besondere Anforderungen mit sich, die an Modellierungssoftware für den MTT gestellt werden müssen. So benötigen sie vor allem Unterstützung bei der Anwendung einer Modellierungsnotation und müssen sich zudem eine gemeinsame Benutzungsoberfläche teilen. Das Modellieren stellt dabei eine spezielle Art von Aufgabe dar, bei dem das Ergebnis zwar durch Text angereichert ist, jedoch vorrangig in einem großen Diagramm besteht. Anstatt immer wieder neue Lösungsansätze für Benutzungsoberflächen zu erarbeiten, bieten HCI-Patterns bewährte Musterlösungen im Bereich der Mensch-Computer-Interaktion. Ziel dieses Papiers ist es, eine Auswahl an HCI-Patterns für die kollaborative Unternehmensmodellierung am MTT vorzustellen. Der Beitrag widmet sich daher zwei Forschungsfragen: 1) Welche Eigenschaften einer Modellierungssoftware finden Nutzer im Kontext der PUM wichtig? 2) Welche bereits existierenden HCI-Patterns helfen bei der Umsetzung der gewünschten Eigenschaften? Dazu führten wir eine Befragung unter Modellierungsexperten durch und glichen die Ergebnisse mit verschiedenen HCI-Katalogen ab. In Abschnitt 2 stellen wir zunächst Hintergründe zur Unternehmensmodellierung und zu HCI-Patterns vor. Anschließend beschreiben wir in Abschnitt 3 unsere Studie, in der wir Personen, die bereits über Erfahrung in Unternehmensmodellierung verfügen, fragten, welche Merkmale ihnen beim Vergleich mehrerer Modellierungsprogramme am MTT auffallen und wie wichtig diese für sie sind. Um nicht nur offensichtliche Merkmale zu erfassen, lehnten wir uns in der Befragungstechnik an die Repertory-Grid-Methode an und ließen die Teilnehmer verschiedene Modellierungsprogramme in der Gruppe ausprobieren und danach im Einzelgespräch vergleichen. Die daraus gewonnene Liste von Merkmalen der Software wird in Abschnitt 4 präsentiert. Auf Basis dieser Merkmale schlagen wir im selben Abschnitt HCI-Patterns vor, die nach unserer Einschätzung geeignet sind, um die Bedürfnisse der Nutzer zu erfüllen. In Abschnitt 5 diskutieren wir die Ergebnisse, gehen auf Limitationen ein und geben einen Ausblick auf zukünftige Forschung.

2 Hintergrund

2.1 Unternehmensmodellierung

Durch Unternehmensmodellierung werden verschiedene Perspektiven auf ein Unternehmen systematisch analysiert und in Modellen festgehalten. Zu solchen Perspektiven zählen die Geschäfts- und Produktionsprozesse, Organisationsstrukturen, Produkte und Dienstleistungen, IT-Systeme und vieles mehr [Ve02]. Welche Perspektiven relevant sind und welche Ausschnitte von Unternehmen modelliert werden, hängt vom Modellierungszweck ab. Es existieren zahlreiche Ansätze zur Unternehmensmodellierung, wobei die partizipative Unternehmensmodellierung (PUM) [SPS07] für das Forschungsthema dieses Papiers besondere

Bedeutung hat. Die PUM sieht vor, dass Methodenexperten und Fachexperten aus dem Unternehmen gemeinsam an der Erstellung der Modelle beteiligt sind. In einer Untersuchung von [LW12] wurde bereits gezeigt, dass das aktive Einbinden von Fachexperten in den Modellierungsprozess dazu führt, dass sich diese Fachexperten mehr einbringen, beispielsweise durch wiederholte Korrekturen. Darüber hinaus bereitet die Modellierung den Fachexperten mehr Freude, wenn sie direkt einbezogen werden. Während in [LW12] ein Methodenexperte mit nur einem Fachexperten modelliert, geht der partizipative Ansatz noch weiter. Kleine Teams von vier bis acht Stakeholdern aus dem Unternehmen erarbeiten in Diskussionen eine gemeinsame Sichtweise, die sie schließlich im Modell dokumentieren [SP18]. Die Stakeholder stellen Fachexperten dar, die über wichtiges Wissen aus ihren jeweiligen Unternehmensbereichen verfügen, das für den Modellierungszweck erforderlich ist. Von Beginn an werden also Fachexperten einbezogen, die wertvolles Wissen in die Modelle einbringen, aber oft auch verschiedene Sichtweisen vertreten, über die es einen Konsens zu erzielen gilt. Damit soll die PUM zu einer höheren Qualität und einer größeren Akzeptanz der Modelle führen. Damit sich die Fachexperten bei der Modellierung auf den Inhalt konzentrieren können, werden sie von Methodenexperten unterstützt. Diese kommen meist von außerhalb des Unternehmens. Ein Methodenexperte leitet als Moderator die Modellierungssitzungen und wird oft durch andere Methodenexperten unterstützt, wie z.B. einen Tool Operator zur Unterstützung bei der Bedienung des Modellierungswerkzeuges oder einen Protokollanten, der Ergänzungen zum Modell und Entscheidungen während des Modellierungsprozesses dokumentiert [Sa14, SP18]. Für den MTT als mögliches Modellierungswerkzeug in der PUM spielt die Modellierungssoftware eine tragende Rolle. Sie muss ermöglichen, dass sich auch die oben erwähnten Fachexperten ohne Vorkenntnisse in der Modellierung möglichst gut zurechtfinden. Die funktionalen Anforderungen umfassen dabei im Wesentlichen das Erstellen der Symbole (geometrischer Formen), die bestimmte Konzepte der Modellierungssprache repräsentieren. Diese werden in der Regel durch Konnektoren (z.B. Pfeile oder Linien) verbunden, um Beziehungen darzustellen. Modellelemente werden meist durch Beschriftungen näher spezifiziert. Wird eine formale Modellierungssprache genutzt, so schreibt diese syntaktische und semantische Regeln vor [SP18].

2.2 HCI-Patterns

HCI-Patterns, auch HCI-Designmuster genannt, beschreiben bewährte Lösungen (Best Practices) für immer wieder auftretende User Interface (UI)-Designprobleme und beeinflussen damit implizit auch die Benutzerfreundlichkeit von Softwaretools [SW07]. Diese Patterns sollen zum einen beim Entwurf von Benutzungsoberflächen unterstützen und zum anderen vermeiden helfen, dass Designer die gleichen Lösungen immer wieder neu „erfinden“. Bewährte Designlösungen werden in Form der Patterns verallgemeinert, wiederverwendbar dokumentiert, und können bei der Lösung ähnlicher Probleme verwendet werden [Wu09]. Die Entwicklung von Benutzungsoberflächen ist ein komplexer Prozess, bei dem die Wiederverwendung von Wissen und damit auch HCI-Patterns den Designern

und Entwicklern dabei hilft, effizienter zu arbeiten und die Produktivität zu verbessern [Bo01, KH10b, KH10a, Gu07].

Ein Pattern besteht im Kern aus Kontext, Problem und Lösung [Al77]. Das HCI-Pattern Breadcrumbs wird beispielsweise im Kontext großer hierarchisch strukturierter Websites eingesetzt. Nutzer sollen sich in dieser Struktur orientieren können (Problem), weshalb ihnen ein Pfad beginnend von der höchsten Hierarchieebene bis zur aktuellen Seite angezeigt wird, wobei jede Ebene im Pfad angesteuert werden kann [va08]. Ursprünglich wurde das Konzept der Patterns und deren Verwendung von Christopher Alexander für Architekturentwürfe entwickelt [Al77]. Die „Gang of Four“(GoF) übernahm das Pattern-Konzept für den Entwurf objektorientierter Software [Ga11], bevor es auch von der HCI-Community übernommen wurde. Während GoF-Patterns auch Hinweise beinhalten, wie man ein Pattern implementiert, adressieren HCI-Patterns allgemeine Designaspekte einer Benutzeroberfläche und ihren Zweck für den Benutzer.

Das Konzept der Patterns beinhaltet nicht nur die Muster selbst, sondern auch die sogenannte Pattern-Sprache. Eine Pattern-Sprache besteht aus Patterns und deren Beziehungen, dargestellt als Netzwerk von Patterns. Übergeordnete Patterns in diesem Netzwerk können durch Patterns auf der darunterliegenden Ebene konkretisiert werden [KH10a]. Da die Patterns einer Pattern-Sprache miteinander verbunden sind, ist es offensichtlich, dass eine Pattern-Sprache Patterns für eine bestimmte Familie von Designproblemen in einer definierten Domäne kombiniert [KH10b, LS04]. Nachfolger- und Vorgängerbeziehungen zwischen (übergeordneten und darunterliegenden) Patterns sind ein Schlüsselkonzept bei der Arbeit mit Pattern-Sprachen, da sie das Auffinden eng verwandter Muster ermöglichen [KH10b].

3 Forschungsmethodisches Vorgehen

3.1 Befragung

Um eine Auswahl passender HCI-Patterns für die PUM an einem MTT geben zu können, befragten wir Modellierungsexperten, welche Eigenschaften ihnen an existierenden Modellierungseeditoren auffallen und welche sie in dieser spezifischen Situation als wichtig erachten. Besondere Eigenschaften dieser Situation sind, dass a) es sich um das Erstellen von Unternehmensmodellen handelt b) mit einem MTT, c) gemeinsam in der Gruppe gearbeitet wird, und d) die Nutzer nicht notwendigerweise über Modellierungserfahrung verfügen.

3.1.1 Stichprobe

An der Befragung nahmen insgesamt acht Personen teil. In der ersten Befragungsrunde nahmen nur zwei Personen teil. Um für die Erkundung der Software die Bedingung der

Gruppenarbeit herzustellen, nahm eine dritte Person aus dem Forschungsteam teil, die aber nicht in die Befragung einging. Darüber hinaus führten wir die Untersuchung mit zwei weiteren Gruppen durch, jeweils bestehend aus drei Probanden. Alle Teilnehmer hatten Vorkenntnisse in Unternehmensmodellierung und stellten damit für uns Experten dar, die die speziellen Nutzerbedürfnisse in diesem Kontext einschätzen können. Sechs der Probanden hatten bereits durch ihre studentische Ausbildung Hintergrundwissen und praktische Erfahrungen mit Unternehmensmodellierung, die anderen zwei Probanden bilden selbst Studenten in Unternehmensmodellierung aus und hatten bereits an partizipativen Modellierungssitzungen als Methodenexperten teilgenommen.

3.1.2 Erkundung verschiedener Modellierungsprogramme

Für die Befragung baten wir die Probanden, in Gruppen von drei Personen am MTT ein Zielmodell zu erstellen. Zielmodelle stellen sehr grundlegende Modelle dar, für die eine gemeinsam erarbeitete Sichtweise auf Ziele und Probleme eines Unternehmens besonders wichtig und die PUM daher besonders geeignet ist. Durch diese praktische Aufgabe wollten wir den Probanden eine konkrete Vorstellung von dieser Modellierungssituation ermöglichen. Wir wählten für die Untersuchung vier Modellierungsprogramme aus, mit denen eine Zielmodellierung möglich ist: 1) das 4EM-Tool, das speziell für die 4EM-Methode entwickelt wurde und mit dem prinzipiell sämtliche in der Methode enthaltene Modelle erstellt werden können, 2) eine von uns entwickelte Software ModelToGather, mit der Zielmodelle nach der 4EM-Notation erstellt werden können sowie 3) Enterprise Studio und 4) Archi, mit denen Modelle nach der Archimate-Notation erstellt werden können. Die Probandengruppen sollten die Softwareprogramme erkunden können, um diese danach besser beurteilen zu können. Dies war vor allem dann nützlich, wenn ein Proband eine Software vorher nicht kannte oder jetzt nach langer Zeit erst wieder damit in Kontakt kam. Im ersten Testdurchlauf stellten wir fest, dass ein Vergleich von mehr als drei Programmen zu viel Zeit in Anspruch nahm. Daher wurden im ersten Durchlauf nur Archi, Enterprise Studio und ModelToGather betrachtet. Im zweiten und dritten Versuch ersetzten wir Archi durch das 4EM-Tool, da sich Archi und Enterprise Studio sehr stark ähneln und wir möglichst unterschiedliche Programme nutzen wollten. Die Erstellung von drei kompletten Modellen während eines Versuchs war zeitlich ebenfalls nicht realisierbar. Daher bereiteten wir Modelle vor, die die Probanden editieren sollten. Um die für die Zielmodellierung benötigten Funktionen abzudecken, führten wir vorher eine Aufgabenanalyse [St06] durch und identifizierten die wesentlichen Aufgaben, die bei einer Zielmodellierung ausgeführt werden. So mussten Komponenten und Relationen erstellt, geändert und gelöscht werden. Die Probanden starteten immer mit einem Initialmodell, das bei jeder Software ein anderes Beispiel enthielt. Sie erhielten einen Zettel, auf dem gekennzeichnet war, welche konkreten Änderungen vorzunehmen waren.

3.1.3 Befragungstechnik

Im Anschluss an die Erkundung der Software führten wir Einzelinterviews durch, bei denen wir uns an der Repertory-Grid-Technik [Ke] orientierten. Dabei sollten die Probanden immer drei Modellierungsprogramme vergleichen, wobei die besondere Fragestellung lautete: Was haben zwei der Programme gemeinsam, wodurch sie sich von dem dritten unterscheiden? Diese etwas kompliziertere Fragestellung soll zu intensiverem Nachdenken anregen und bestenfalls implizites Wissen zugänglich machen [Le01, Ei93]. Die Methode sieht normalerweise eine größere Anzahl zu vergleichender Elemente vor, die nur mit einer entsprechend großen Stichprobe abgedeckt werden kann. Aus Effizienzgründen entschieden wir uns für eine Beschränkung auf drei Programme. Nachdem wir die Liste dieser Merkmale gemeinsam mit dem Probanden erstellt hatten, sollte dieser einschätzen, wie wichtig er jedes Merkmal einschätzt für die Zusammenarbeit am Multi-Touch-Tisch auf einer Skala von eins (unwichtig) bis fünf (sehr wichtig) [Le01, Ei93]. Die von den Probanden genannten Merkmale der Modellierungssoftware wurden induktiv zu Kategorien zusammengefasst analog zu einer qualitativen Inhaltsanalyse [DB16].

3.2 Vorgehen zur Auswahl von HCI-Patterns

Für die Recherche über HCI-Patterns nutzten wir als Grundlage vielzitierte Pattern-Sammlungen [Ti11, Ti99, va08] sowie weitere Pattern-Sammlungen, die für den betrachteten Kontext passend waren [La03, LHS13, CL16]. Remy et al. bieten darüber hinaus eine Sammlung von HCI-Patterns speziell für den MTT [Re10]. Durch eine vorherige Beobachtungsstudie, vorgestellt in [GSSck], formulierten wir bereits Kategorien, die groben Konzepten entsprechen, denen untergeordnete konkrete HCI Patterns dienen. In jener Studie selektierten wir bereits auf Grundlage der Analyse einer konkreten Software sowie Beobachtungsdaten Patterns, die potentiell in Modellierungssoftware zum Einsatz kommen sollten. Die hier vorgestellte Untersuchung soll unsere bisherigen Erkenntnisse nun erweitern, indem eine weitere Forschungsmethode, die Befragung von Modellierungsexperten, angewandt wird. Deshalb übernahmen wir aus dem oben genannten Beitrag Inhaltskategorien zur Einordnung unserer neuen Befragungsergebnisse. In der Befragung ermittelten wir Softwaremerkmale, aus denen wir HCI-Patterns ableiteten und diese dann den Inhaltskategorien zuordneten. Die Kategorien helfen bei der Gestaltung der Pattern-Sprache, in der Zusammenhänge zwischen den Patterns dargestellt werden. Die erste Kategorie bezieht sich auf das Einsparen bzw. effiziente Nutzen von **Platz**, da wir mit dem MTT nur über eine begrenzte Arbeitsfläche verfügen. Als zweite Kategorie definierten wir **visuelle und physische Erreichbarkeit**, da sichergestellt werden soll, dass jeder Modellierungsteilnehmer alle Elemente eines Modells immer gut erkennen und möglichst auch für Eingaben erreichen soll. Mit der dritten Kategorie, **ausgewogene Partizipation**, betonen wir das Ziel, dass sich alle an einer Modellierungssitzung in gleichem Maße einbringen können. Die Kategorie **Modellierungsunterstützung** fasst Bedürfnisse zusammen, die speziell mit der Modellierungstätigkeit zusammenhängen. Weiterhin definierten wir ein **intuitives Interface**

sowie **einfache Handhabung** jeweils als Kategorie, wobei letztere sich darauf bezieht, dass Interaktionen mit der Software so einfach und wenig aufwendig wie möglich gestaltet werden sollen. Die Kategorie **Input Devices** fasst die Möglichkeiten für Nutzereingaben am MTT zusammen. Die letzte Kategorie **Rationale dokumentieren** spiegelt das Bedürfnis danach wider, Entscheidungsprozesse einschließlich der Urheberschaft von Vorschlägen und Ideen nachvollziehen zu können. Da wir uns am Prinzip einer Pattern-Sprache orientieren, kann ein genanntes Merkmal auch mehr als einer Oberkategorie zugeordnet und damit auch dieser dienlich sein. Auf diese Weise werden die Merkmale in die Pattern-Sprache integriert.

4 Ergebnisse

Die in der Befragung aufgezählten Merkmale der Software entsprachen teilweise bereits konkreten Patterns, teilweise lassen sie sich aber auch als Konzepte auf einer Ebene zwischen den Oberkategorien und konkreten Patterns einordnen. In den folgenden Abschnitten präsentieren wir die Ergebnisse der Befragung und zeigen passende konkrete Patterns dazu auf, die wir den zuvor erwähnten Katalogen entnommen haben. Um einen Eindruck von der resultierenden Pattern-Sprache zu vermitteln, verweisen wir auf Abbildung 1, wobei die konkreten Patterns, die wir den Pattern-Katalogen entnommen haben, unterstrichen dargestellt sind. Pfeile zwischen den Patterns stellen dar, welche Patterns Bestandteil zur Umsetzung eines anderen Patterns sein können.

Eingeschränkter Platz: Vier Befragte gaben als Merkmal für Modellierungssoftware an, dass Elemente entweder problemlos angeordnet waren oder aufgrund des Platzmangels stark überlappten. Dabei wurden allerdings unterschiedliche Aspekte benannt: Das problemlose Anordnen der Komponenten wie z.B. Ziele ohne Überlappung wurde einmal genannt und mit dem Wert 5 (von fünf Stufen) als wichtig eingeschätzt. Zwei Befragte gaben mit einer Wichtigkeit von 3 bzw. 4 an, dass Elemente sich allgemein nicht überlappen, während ein Befragter mit einer Wichtigkeit von 2 nannte, dass Menüs nicht verdeckt sein sollten. Dieses Problem sollte nicht unterschätzt werden, da Unternehmensmodelle sehr groß werden können. Tidwell [Ti11] und van Welie [va08] schlagen **Collapsible Panels** vor, um Platz zu sparen, also Bedienfelder, die beliebig ein- und ausgeklappt werden können, je nach Bedarf. Darüber hinaus findet man bei Tidwell [Ti11] und van Welie [va08] als Pattern den **Drop Down Chooser**. Auf Antippen klappt sich dabei nach unten ein Menü auf, aus dem man eine Option wählen kann. **Hover Tools** [Ti11] werden eigentlich in der Mausinteraktion genutzt. Schwebt der Mauszeiger über einem Element, können dazugehörige Informationen eingeblendet werden. Diese Möglichkeit gibt es bei Touch-Interaktion nicht. Dennoch könnten Zusatzelemente, die nicht dauerhaft benötigt werden, bei Berührung für eine kurze Dauer eingeblendet werden. Es spart ebenfalls Platz, wenn Schaltflächen nur mit **Symbolen und Icons** beschriftet sind [Ti11, va08, LHS13]. Eine weitere Lösung, um dem Platzmangel entgegenzuwirken, ist der Einsatz eines größeren MTT wie bei dem **Large Collaboration**

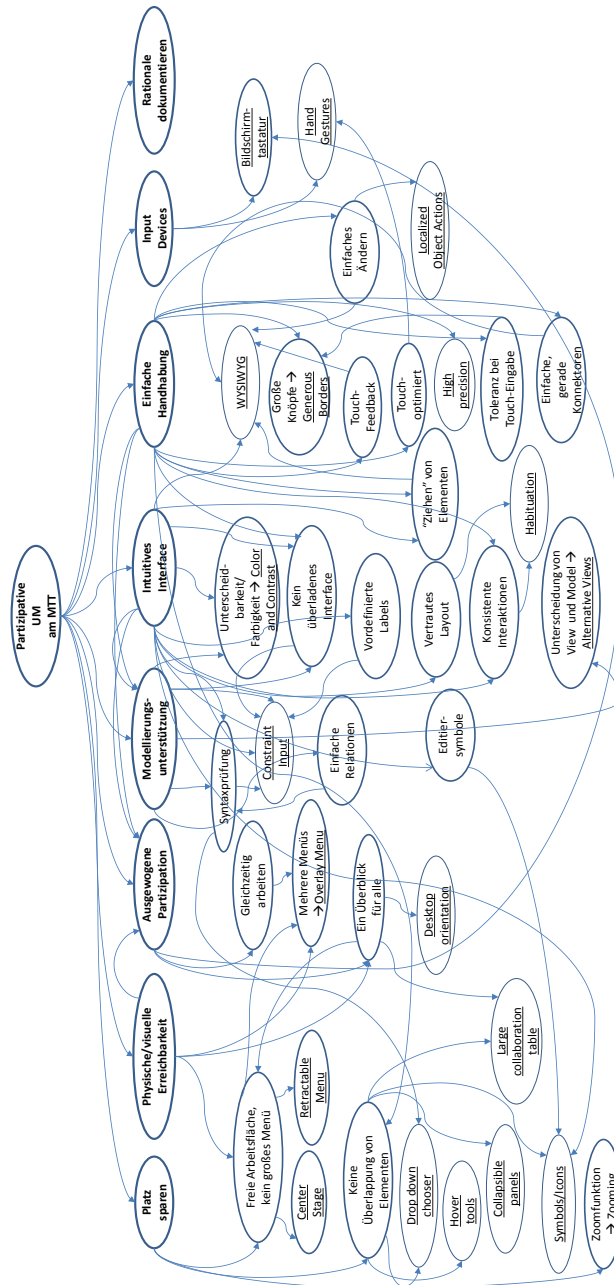


Abb. 1: HCI-Patterns für kollaborative Unternehmensmodellierung am MTT eingeordnet in eine Pattern-Sprache; konkrete Patterns aus Pattern-Katalogen wurden unterstrichen.

Table aus der Pattern-Sammlung von [Re10].

Eine **Zoom-Funktion** wurde als Software-Merkmal von drei Befragten genannt und mit einer durchschnittlichen Wichtigkeit von 3,67 bewertet. Sie entspricht einem HCI-Pattern, das in [Ti11] und [Re10] beschrieben wird. Damit könnten auch größere Modelle auf einer begrenzten Arbeitsfläche erstellt werden.

Eine freie Arbeitsfläche ohne ein fixes großes Hauptmenü, das Platz wegnimmt, nannten drei Personen und bewerteten dies mit einer Wichtigkeit von durchschnittlich 3,67. Viele Programme nutzen bereits das Pattern **Center Stage** [Ti11, va08], bei dem eine große Fläche für den Hauptzweck eines Programmes im Zentrum reserviert ist, während sekundäre Elemente weniger Platz zugeteilt bekommen und in kleineren Panels am Rand angeordnet sind. Dies ist möglicherweise jedoch bei großen Unternehmensmodellen nicht ausreichend. Van Welie führt in seiner Pattern-Liste **Retractable Menus** auf, wonach das Hauptmenü einklappbar sein soll, wenn besonders viel Platz benötigt wird [va08]. **Overlay Menus** werden an der Stelle geöffnet, an der der Mauszeiger [va08] bzw. der Touch-Punkt liegt.

Physische und visuelle Erreichbarkeit: Die freie Arbeitsfläche ohne fixes Menü dient auch der physischen Erreichbarkeit. Gibt es nur ein Hauptmenü, so bedeutet dies meist, dass nur die Personen das Menü erreichen, die diesem am nächsten stehen. Als Lösung dieses Problems kommt vor allem das **Overlay Menu** in Frage, welches an jeder gewünschten Stelle geöffnet werden kann. Dies wurde auch von zwei Personen genannt, allerdings mit einer durchschnittlichen Wichtigkeit von 2,5 bewertet. Dass eine Modellierungssoftware einen Überblick für alle bietet, wurde von einer Person genannt mit einer Wichtigkeit von 4. Hier empfehlen sich die Patterns **Center Stage** und **Desktop Orientation** an. Mit Desktop Orientation werden Inhalte auf einen jeweiligen Betrachter ausgerichtet, so dass jeder Nutzer die Chance hat, das Modell aus seiner Position „richtig herum“ zu sehen.

Ausgewogene Partizipation: Sieben Befragte nannten gleichzeitiges Arbeiten als Merkmal und gaben durchschnittlich eine Wichtigkeit von 3,71 an. Das heißt, es war den Probanden tendenziell wichtig, dass sie mit mehreren Personen parallel mit dem MTT interagierten. Eng damit zusammen hängt das Merkmal, dass mehrere Elemente zur gleichen Zeit bearbeitet werden können, wozu wir drei Nennungen mit einer durchschnittlichen Wichtigkeit von 4,33 dokumentieren konnten. Remy et al. [Re10] nennen hier das allgemeine Pattern **Balanced Participation**, bei dem sich Nutzer, die in einer Gruppe arbeiten, möglichst alle einbringen können. Konkret unterstützend für ausgewogene Partizipation schätzen wir die zuvor erwähnten **Overlay Menus** ein, die mehrfach erzeugt werden können. Ausgewogene Partizipation wird ebenfalls durch das Merkmal des Überblicks für alle unterstützt. Hier bietet sich erneut das Pattern **Desktop Orientation** an. Auch ein großer MTT (**Large Collaboration Table**, [Re10]) trägt zu einem guten Überblick bei.

Modellierungsunterstützung: Drei Befragte nannten Syntaxprüfung als ein Merkmal der Software, wobei sie dem eine Wichtigkeit von durchschnittlich 2 zusprachen. Eine der Personen bewertete es mit 2, wenn korrekte und inkorrekte Syntax angezeigt würden. Vordefinierte Labels können ebenfalls beim Modellieren unterstützen. Eine Person erwähnte mit der Wichtigkeit 4 das einfache Beschreiben von Relationen zwischen Komponenten und mit gleicher Wichtigkeit die Nutzung von Popups. Ein Befragter nannte vordefinierte Typen für Relationen, zwei Befragte vordefinierte Typen für Komponenten. In allen Fällen wurde eine Wichtigkeit von 4 angegeben. Als hilfreiche HCI-Patterns identifizierten wir hier **Constraint Input** [va08] sowie **Drop Down Chooser** [va08, Ti11, Ti99]. Beispielsweise könnte die Software zur Laufzeit prüfen, welche Relationslabels für die Beziehung zwischen zwei Komponenten gerade zulässig sind. Soll eine neue Komponente erstellt werden, so könnte dies über ein Drop-Down-Menü geschehen, das nur die Komponententypen enthält, die das Modell vorsieht.

Einige genannte Merkmale sind abhängig von der Modellierungssprache. So scheinen Farben (fünf Nennungen, Wichtigkeit durchschnittlich 4) und allgemein die Unterscheidbarkeit der Modellkomponenten (eine Nennung, Wichtigkeit 4) wichtige Merkmale zu sein. Passendes Pattern ist hier **Color and Contrast** [LHS13]. Darüber hinaus wurden einfache Relationen als Merkmal zweimal genannt mit einer Wichtigkeit von durchschnittlich 4. Bei den möglichen Labels für eine Relation sollte es nicht zu viele Auswahlmöglichkeiten geben, die die Nutzer überfordern. Dies liegt jedoch in der Sprache begründet. Allerdings könnte durch eine Syntaxprüfung zur Laufzeit die Anzahl der Labels reduziert und die Auswahl damit erleichtert werden.

Sieben Befragte nannten ein nicht überforderndes im Gegensatz zu einem überladenen Interface und gaben durchschnittlich eine Wichtigkeit von 3,43 an. Ein nicht überladenes Interface macht es intuitiver und unterstützt ebenfalls das Modellieren. Im Zusammenhang damit wurde auch ein Fokus auf wesentliche Komponenten von zwei Personen genannt mit einer Wichtigkeit von durchschnittlich 3. Sowohl Archimate als auch 4EM sehen vor, dass die verschiedenen Modelle, die mit diesen Notationen angefertigt werden können, miteinander verbunden werden können. In einigen Editoren ist es möglich, nicht nur die Elemente der aktuellen Modellart zu sehen, sondern den gesamten Umfang der Notationselemente. Auch hier könnte das Pattern **Constraint Input** eingesetzt werden. Dem gegenüber stehen jedoch vier Nennungen, die sich auf das Anzeigen aller möglichen Komponenten beziehen, allerdings mit einer Wichtigkeit von durchschnittlich lediglich 2,5. Zwei Personen nannten als Merkmal, dass die Software viele Funktionen anbietet im Gegensatz zu einer kleinen Auswahl von Basisfunktionen, bewerteten dies aber mit einer Wichtigkeit von durchschnittlich nur 1,5. In unserer Stichprobe scheint damit eher ein Wunsch nach Reduktion der Auswahlmöglichkeiten vorzuherrschen.

In einigen Modellierungsprogrammen ist es üblich, zwischen „Ansicht“ (View) und „Modell“ zu unterscheiden. Wird in diesem Falle eine Komponente in der Ansicht gelöscht, so verschwindet sie dort zwar, ist aber noch im Modell enthalten. Dies wurde von einer Person genannt und mit dem Wert 2 als weniger wichtig eingeschätzt. Bei [Ti11] findet man das Pattern **Alternative Views** für verschiedene Ansichten.

Intuitives Nutzer-Interface: Sieben Personen nannten intuitive Bedienbarkeit als Merkmal einer Modellierungssoftware mit einer durchschnittlichen Wichtigkeit von 4,7. Viele der in der vorherigen Kategorie zur Modellierungsunterstützung genannten Merkmale und Patterns beziehen sich auch auf die intuitive Bedienung der Software. Dazu zählen eine nicht überladene Bedienoberfläche, Unterscheidbarkeit und Farben der Modellelemente und vordefinierte Labels. Dementsprechend sind für intuitive Bedienbarkeit auch dieselben Patterns nützlich, vor allem **Constraint Input** und **Color and Contrast**. Auch die allgemeine Syntaxprüfung macht die Bedienung intuitiver.

Ein vertrautes Layout, das von anderen Anwendungen bekannt ist, wurde von einem Befragten mit der Wichtigkeit 2 angegeben. Tidwell [Ti11] beschreibt ein Muster für Nutzerverhalten, das sie Habituation nennt. Es beschreibt, dass Nutzer erwarten, dass bestimmte Interaktionen so funktionieren, wie sie es von anderen Programmen gewohnt sind. Dass Interaktionen für ähnliche Aktionen gleich und damit konsistent über die gesamte Anwendung, insbesondere das Öffnen von Menüs, gleich sein sollten, erwähnte eine Person mit der Wichtigkeit 1. Auch dies setzen wir in Zusammenhang mit dem Bedürfnis nach **Habituation**, wenngleich dies für die Probanden von geringerer Bedeutung zu sein schien. Auch das „Herüberziehen“ von Objekten, wenn z.B. eine Komponente erzeugt (eine Nennung, Wichtigkeit 4) oder eine Relation von einer zur anderen Komponente gezeichnet wird (Vier Nennungen, durchschnittliche Wichtigkeit 3,75), zählt zur intuitiven Bedienung. Dies entspricht der Idee des Patterns **WYSIWYG** (What you see is what you get) [Ti99], bei dem der Nutzer in Echtzeit das Resultat seiner Interaktionen mit dem Modell sehen kann. Eine Person erwähnte Symbole, die die Möglichkeiten des Editierens signalisieren, mit einer Wichtigkeit von 3, was durch das Pattern **Symbols/Icons** umgesetzt wird.

Einfache Handhabung: Wie bereits angedeutet, gibt es einige Überschneidungen zwischen intuitiver Bedienbarkeit und einfacher Handhabung. Dazu zählen eine nicht überladene Bedienoberfläche, das „Ziehen“ von Objekten und konsistente Interaktionen bei Menüs. Entsprechend sollten die oben bereits erwähnten Patterns hilfreich sein. Auch sollten sich Elemente nicht zu stark überlappen.

Darüber hinaus sollten nach der Meinung von zwei Personen mit einer durchschnittlichen Wichtigkeit von 3,5 Knöpfe groß sein. Bei drei Befragten fanden wir sehr unterschiedliche Aussagen zur Präzision von Touch-Eingaben bzw. zur Toleranz bei „vorschnellen“ Eingaben. Einerseits kann es frustrierend sein, wenn eine Software schwerfällig oder gar nicht auf Eingaben reagiert. Andererseits kann es ebenso den Arbeitsablauf stören, wenn das Nutzer-Interface auf jede Berührung übersensibel reagiert. Das Pattern **Generous Borders** [Ti11] entspricht dem Bedürfnis nach Knöpfen, die groß genug sind, und sorgt für eine Toleranz bei Eingaben durch größere Abstände zwischen Knöpfen. Remy et al. [Re10] führen das Pattern **High Precision Input** an, unter dem sie verschiedene Maßnahmen zusammenfassen, um Touch-Eingaben möglichst genau zu erfassen, z.B. den Einsatz von Gesten oder Algorithmen, die die Präzision der Eingabeerfassung verbessern.

Zwei Personen nannten mit einer durchschnittlichen Wichtigkeit von 2,5 Touch-Feedback. Das Nutzer-Interface soll entweder optische oder haptische Rückmeldung bei Berührung

geben. Dies entspricht der Idee des **WYSIWYG**. Drei Personen bewerteten durchschnittlich mit dem Wert 3, dass die Software für Touch-Eingaben ausgelegt oder optimiert sein sollte. In diesem Zusammenhang steht möglicherweise auch eine Aussage in unserer Befragung über ein eigenes Bedienkonzept, das mit einer Wichtigkeit von 3 eingeschätzt wurde. Wenn eine Software am MTT genutzt wird, sollte sie auch Touch-Gesten nutzen (**Hand Gestures**, [Re10]).

Ein weiteres Merkmal ist das einfache Ändern des Modells. Sechs Personen gaben mit einer durchschnittlichen Wichtigkeit von 4 die Möglichkeit zur Änderung des Komponententyps an. Im Gegensatz dazu müsste ein Nutzer eine Komponente löschen, neu erstellen und eventuell dazugehörige Relationen aktualisieren. Auch das Ändern von Relationen wurde von drei Personen mit einer durchschnittlichen Wichtigkeit von 4 genannt. Für diese Änderungen erscheinen **Localized Object Actions** [Ti99] geeignet. Sie sehen vor, dass Bedienelemente zum Bearbeiten eines Objektes gruppiert und in der Nähe dieses Objektes angeordnet werden. Zwei Nennungen bezogen sich allgemein auf nutzerfreundliche Änderbarkeit mit einer Wichtigkeit von durchschnittlich 3,5. Bei drei Nennungen ging es um das schnelle Editieren von Bezeichnungen ohne vorherige umständliche oder unklare Interaktionen, was mit einer durchschnittlichen Wichtigkeit von 2,3 bewertet wurde. Eine Umsetzung des Patterns **WYSIWYG** wäre, dass, ohne sich durch Menüs oder andere zusätzliche Schritte arbeiten zu müssen, ein Textfeld sofort geändert werden könnte.

Ein Befragter störte sich an Konnektoren, die nicht gerade sondern kompliziert zwischen Komponenten verliefen. Mit einer Wichtigkeit von 4 bewertete der Befragte das Merkmal einfacher gerader Konnektoren. Hier müsste sich eine Software stärker am Pattern **WYSIWYG** orientieren, damit der Verlauf der Konnektoren für die Nutzer berechenbarer ist.

Auch das Herüberziehen von Komponenten und Konnektoren unterstützt die einfache Handhabbarkeit und setzt, wie oben bereits angeführt, das Pattern **WYSIWYG** um.

Input Devices: Vier Personen erwähnten die Bildschirmtastatur als Merkmal einer Modellierungssoftware und bewerteten dies mit einer Wichtigkeit von durchschnittlich 4,5. Sofern diese erreichbar gemacht wird für alle, z.B. durch die Möglichkeit mehrere Instanzen, die beliebig platziert werden können, dient dies auch der ausgewogenen Partizipation. Die genannten Merkmale, dass eine Anwendung für Touch optimiert ist oder ein eigenes Bedienkonzept hat, passen zum Pattern **Hand Gestures**.

In den Aussagen der Probanden fanden wir keine Hinweise zur Kategorie „Rationale dokumentieren“.

5 Diskussion

5.1 Zusammenfassung

Ein Pattern oder Muster ist erst dann ein Muster, wenn es sich um eine wiederholt eingesetzte Lösung für ein wiederkehrendes Problem handelt [Gal1]. Dies gilt auch im Bereich der

Design-Lösungen für Benutzungsoberflächen. In diesem Papier haben wir eine Studie vorgestellt, in der nach Patterns für Software gesucht wurde, die speziell in der partizipativen und kollaborativen Unternehmensmodellierung am MTT eingesetzt werden soll. Ziel war es also nicht, neue Patterns zu entwickeln, sondern aus vorhandenen Pattern-Sammlungen diejenigen herauszusuchen, die die besonderen Nutzerbedürfnisse in diesem speziellen Kontext erfüllen helfen. Um ein schärferes Bild einer Modellierungssoftware und auch von den Nutzerbedürfnissen zu erhalten, nutzten wir eine Befragungstechnik angelehnt an die Repertory-Grid-Technik. Im Vergleich von verschiedenen Modellierungsprogrammen nannten uns die Probanden Merkmale entsprechender Software. Einige dieser Merkmale lassen sich bereits als Muster in Pattern-Sammlungen finden, wie z.B. Zooming, Color and Contrast oder Symbole. Andere Merkmale wurden eher als allgemeine Eigenschaften oder Anforderungen formuliert, die wir als Konzepte in eine Pattern-Sprache einordneten, wie z.B. das Bedürfnis, dass sich Elemente nicht überlappen sollten, oder das Bedürfnis, dass Teile des Modells möglichst einfach geändert werden können. Diese Merkmale brachten wir mit konkreten Patterns in Verbindung, die wir für die Erfüllung der dazugehörigen Bedürfnisse für passend halten.

Bei genauerer Betrachtung der Patterns fällt auf, dass es zahlreiche Patterns gibt, die positiv zusammenwirken oder einander unterstützen. So ist Desktop Orientation, also das flexible Ausrichten des Inhalts auf wechselnde Betrachter, auch eine Möglichkeit, visuelle Erreichbarkeit für alle zu gewährleisten. Allerdings ergeben sich auch zahlreiche Widersprüche zwischen Patterns. So kann die Desktop Orientation die Zusammenarbeit und damit auch die ausgewogene Partizipation stören. Die Nutzer müssen sich einig sein über eine Veränderung der Ausrichtung. Dabei kann es zu Unterbrechungen im Arbeitsfluss kommen und ohne klare Organisation können bestimmte Ausrichtungen und damit auch Nutzer blockiert werden. Das Overlay Menu wurde als Möglichkeit vorgeschlagen, um große Hauptmenüs zu ersetzen und damit Platz zu sparen und ausgewogene Partizipation zu unterstützen. Werden allerdings mehrere Menüs gleichzeitig genutzt, die sehr groß sind, kann dies wieder Platz wegnehmen.

Einige der von den Befragten genannten Merkmale sind auf die Modellierungssprache zurückzuführen, wie z.B. der Einsatz von Farben, welcher von mehr als der Hälfte der Probanden tendenziell als wichtig eingestuft wurde, oder die Anzahl zur Verfügung stehender Labels, die potentiell überfordern kann. In der Praxis sollte vor Durchführung eines Modellierungsprojektes auf diese Aspekte geachtet werden, wenn eine Modellierungssprache ausgesucht wird.

In den Interviews wurde deutlich, dass Platz auf der Arbeitsfläche sowie intuitive Bedienbarkeit, einfache Handhabung und physische und visuelle Erreichbarkeit unabdingbare Voraussetzungen für ausgewogene Partizipation und Modellierung sind. Obwohl nach der Wichtigkeit der genannten Merkmale für die Zusammenarbeit gefragt wurde und sie mit Zusammenarbeit eigentlich nicht direkt in Zusammenhang stehen, wurden besonders diese Kategorien häufig genannt. Sie stellen damit eine Art Hygienefaktoren für die gemeinsame Modellierung dar. Fehlen sie, ist Zusammenarbeit nicht möglich, ihre besondere Förderung

steigert die Qualität der Zusammenarbeit an sich vermutlich jedoch nicht. Die Software muss jedoch darüber hinausgehende Merkmale aufweisen, um dann tatsächlich Zusammenarbeit zu fördern. So wurde mit tendenziell hoher Wichtigkeit gleichzeitiges Arbeiten bewertet. Ein Pattern, das das gleichzeitige Bearbeiten mehrerer Elemente beschreibt, haben wir in den Pattern-Sammlungen bisher nicht gefunden. Eventuell könnte sich hier in Zukunft ein neues Pattern formulieren lassen, welches das eher unspezifische Pattern *Balanced Participation* von [Re10] etwas konkreter fortführt.

5.2 Limitationen und Ausblick

Bei der vorgestellten Studie handelt es sich nur um eine kleine Untersuchung mit wenigen Probanden und wenigen Modellierungsprogrammen. Eine Ausweitung der Untersuchung wäre wünschenswert. Dabei möchten wir ein breiteres Spektrum von Nutzern einbeziehen. Dazu gehören neben Experten in der Modellierung auch Nichtexperten, sowie Nutzer, die sowohl erfahren als auch unerfahren mit Touch-Interaktionen sind. In einem realistischeren Kontext ergeben sich vielleicht andere Schwerpunkte, wie z.B. die Dokumentation der Entscheidungsfindung oder Urheberschaft (Rationale), die in der aktuellen Befragung keine Rolle spielte. Darüber hinaus streben wir an, mehr Softwareprodukte zu vergleichen, die speziell für MTT ausgerichtet sind, wie z.B. *metasonic Process Touch*. Zukünftige Studien müssten dabei über die bisher betrachtete Zielmodellierung hinaus gehen und andere Modellarten wie z.B. Prozessmodelle einbeziehen. Dennoch liefert uns die Untersuchung bereits einige Erkenntnisse über die Nützlichkeit bestimmter HCI-Patterns. Ob die Liste der Patterns vollständig ist, können wir nicht mit Sicherheit beantworten. Unsere Ergebnisse sind als Hypothesen zu verstehen, die es zu prüfen gilt. Erst mit zunehmender Nutzung von MTT für kollaborative Modellierung und der Untersuchung weiterer Modellierungsprogramme können mehr und mehr zuverlässige Aussagen über wiederholt eingesetzte Musterlösungen getroffen werden. Unser Beitrag stellt daher einen ersten Ansatz zur Entwicklung einer Pattern-Sprache für PUM an MTT dar. Vorausgesetzt, dass diese wie oben beschrieben weiterentwickelt wird, kann sie als Wissens- und Kommunikationsbasis für User-Interface-Entwickler in diesem spezialisierten Bereich dienen.

Literaturverzeichnis

- [AI77] Alexander, Christopher: A pattern language, Jgg. 2 in Center for Environmental Structure series. Oxford Univ. Pr, 1977.
- [Bo01] Borchers, Jan O.: A pattern approach to interaction design. 15:359–376, 2001.
- [CL16] Experiences - A Pattern Language for User Interface Design. <http://www.maplefish.com/todd/papers/Experiences.html#Interaction>.
- [DB16] Döring, Nicola; Bortz, Jürgen: Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften. Springer-Lehrbuch. Springer, 5. vollständig überarbeitete, aktualisierte und erweiterte auflage. Auflage, 2016. Döring, Nicola (VerfasserIn) Bortz, Jürgen (VerfasserIn) Pöschl-Günther, Sandra (MitwirkendeR).

- [Ei93] Einführung in die Repertory Grid-Technik. Bd. 1. Grundlagen und Methoden, 1993. Scheer, Jörn W. (Hrsg.) Catina, Ana (Hrsg.).
- [Ga11] Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John: Design patterns. Addison-Wesley professional computing series. Addison-Wesley, 39. printing. Auflage, 2011.
- [GSSck] Gutschmidt, Anne; Sauer, Valentina; Sandkuhl, Kurt: Identifying HCI Patterns for the Support of Participatory Enterprise Modeling on Multi-Touch Tables. *The Practice of Enterprise Modeling*, in Druck.
- [Gu07] Guerrero-García, Josefina; González-Calleros, Juan Manuel; González-Monfil, Adelaida; Pinto, David: A method to align user interface to workflow allocation patterns. In (González-Calleros, Juan Manuel; Collazos Ordoñez, Cesar A.; Guerrero-García, Josefina, Hrsg.): *Proceedings of the XVIII International Conference on Human Computer Interaction. ICPS. ACM*, S. 1–8, 2007.
- [Gu18a] Gutschmidt, Anne: Empirical Insights into the Appraisal of Tool Support for Participative Enterprise Modeling. In: *Proceedings of the 9th International Workshop on Enterprise Modeling and Information Systems Architectures*, Rostock, Germany, May 24th - 25th, 2018. S. 70–74, 2018.
- [Gu18b] Gutschmidt, Anne: On the Influence of Tools on Collaboration in Participative Enterprise Modeling—An Experimental Comparison between Whiteboard and Multi-Touch Table. In: *ISD 2018*. 2018.
- [Ke] Kelly, George Armstrong: *The psychology of personal constructs*. Routledge. Kelly, George Armstrong (VerfasserIn).
- [KH10a] Kruschitz, Christian; Hitz, Martin: Analyzing the HCI design pattern variety. In (Hanyuda, Eiichi, Hrsg.): *Proceedings of the 1st Asian Conference on Pattern Languages of Programs. ACM*, S. 1, 2010.
- [KH10b] Kruschitz, Christian; Hitz, Martin: *Human-Computer Interaction Design Patterns*. 3, 2010.
- [La03] *User Interface Design Patterns*. <https://www.cs.helsinki.fi/u/salaakso/patterns/index.html>, Accessed: 2019-05-02.
- [Le01] Leach, Chris; Freshwater, Kate; Aldridge, Jan; Sunderland, Joanne: Analysis of repertory grids in clinical practice. 40:225–248, 2001.
- [LHS13] Lockton, Dan; Harrison, David; Stanton, Neville A: Exploring design patterns for sustainable behaviour. *The Design Journal*, 16(4):431–459, 2013.
- [LS04] Lukosch, Stephan; Schümmer, Till: Communicating Design Knowledge with Groupware Technology Patterns. In (de Vreede, Gert-Jan, Hrsg.): *Groupware: Design, Implementation and Use*, Jgg. 3198 in *Lecture Notes in Computer Science*, S. 223–237. Springer Berlin Heidelberg, 2004.
- [LW12] Luebbe, Alexander; Weske, Mathias: Determining the Effect of Tangible Business Process Modeling. In (Plattner, Hasso; Meinel, Christoph; Leifer, Larry, Hrsg.): *Design Thinking Research: Studying Co-Creation in Practice*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 241–257, 2012.

- [MH14] Mercier, E.; Higgins, S.: Creating joint representations of collaborative problem solving with multi-touch technology. *Journal of Computer Assisted Learning*, 30(6):497–510, 2014.
- [PH09] Piper, Anne Marie; Hollan, James D.: Tabletop Displays for Small Group Study: Affordances of Paper and Digital Materials. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09, ACM, New York, NY, USA, S. 1227–1236, 2009.
- [Re10] Remy, Christian; Weiss, Malte; Ziefle, Martina; Borchers, Jan: A Pattern Language for Interactive Tabletops in Collaborative Workspaces. In: *Proceedings of the 15th European Conference on Pattern Languages of Programs*. EuroPLoP '10, S. 9:1–9:48, 2010.
- [Sa14] Sandkuhl, Kurt; Stirna, Janis; Persson, Anne; Wißotzki, Matthias: *Enterprise modeling. The Enterprise Engineering Series*. Springer Berlin Heidelberg, 2014.
- [SP18] Stirna, Janis; Persson, Anne: *Enterprise Modeling - Facilitating the Process and the People*. Springer International Publishing, 2018.
- [SPS07] Stirna, Janis; Persson, Anne; Sandkuhl, Kurt: Participative Enterprise Modeling: Experiences and Recommendations. In (Krogstie, John; Opdahl, Andreas; Sindre, Guttorm, Hrsg.): *Advanced information systems engineering, Lecture Notes in Computer Science* 4495. Springer, 2007.
- [St06] Stanton, Neville A.: Hierarchical task analysis. 37:55–79, 2006. *Journal Article Research Support, Non-U.S. Gov't Review*.
- [SW07] Specker, Markus; Wentzlaff, Ina: Exploring Usability Needs by Human-Computer Interaction Patterns. In (Winckler, Marco; Johnson, Hilary; Palanque, Philippe, Hrsg.): *Task Models and Diagrams for User Interface Design*, Jgg. 4849 in *Lecture Notes in Computer Science*, S. 254–260. Springer Berlin Heidelberg, 2007.
- [Ti99] Common Ground. http://www.mit.edu/~jtidwell/common_ground_onefile.html, Accessed: 2019-05-02.
- [Ti11] Tidwell, Jenifer: *Designing interfaces*. Safari Tech Books Online. O'Reilly, 2nd ed.. Auflage, 2011.
- [va08] Welie.com: *Patterns in Interaction Design*. <http://www.welie.com/patterns/index.php>, Accessed: 2019-07-30.
- [Ve02] Vernadat, F.B.: Enterprise modeling and integration (EMI): Current status and research perspectives. *Annual Reviews in Control*, 26(1):15 – 25, 2002.
- [Wu09] Wurhofer, Daniela; Obrist, Marianna; Beck, Elke; Tscheligi, Manfred: Introducing a Comprehensive Quality Criteria Framework for Validating Patterns. In (Dini, Petre, Hrsg.): *Computation world*. IEEE, S. 242–247, 2009.