

Alfred Zimmermann, Alexander Rossmann (Hrsg.): Digital Enterprise Computing 2015,  
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2015 195

---

# Konzeption und prototypische Umsetzung eines Architekturcockpits

Sebastian Breitbach<sup>1</sup>, Baris Cubukcuoglu<sup>1</sup>, Tobias Gorhan<sup>1</sup>, Daniel Hammer<sup>1</sup>, Dierk Jugel<sup>2</sup> und Christoph Wenzel<sup>1</sup>

**Abstract:** EAM ist ein holistischer Ansatz, um komplexe IT- und Unternehmensstrukturen darzustellen. Dabei ist es von zentraler Bedeutung diese Strukturen möglichst komplett und übersichtlich zu visualisieren. Ein Ansatz dies zu erreichen ist eine multiperspektivische Darstellung von mehreren Views in einem Architekturcockpit. Dabei können mehrere Views simultan betrachtet und analysiert werden. Dadurch ist es möglich die Auswirkungen einer Analyse des Views eines Stakeholders simultan aus den Views anderer Stakeholder betrachten zu können um eventuelle Wechselwirkungen zu erkennen und einen allgemeinen Überblick über die Unternehmensarchitektur zu behalten. In dieser Arbeit zeigen wir, von der Konzeption über die Umsetzung bis zu einem Anwendungsbeispiel, wie ein solches Architekturcockpit realisiert werden kann.

**Keywords:** EAM, Enterprise Architecture Management, Unternehmensarchitektur, Visualisierung, Visuelle Analyse

## 1 Einleitung

Unternehmen sind komplexe und integrierte Systeme, bestehend aus u.a. Geschäftsprozessen, Geschäftseinheiten, Ressourcen und Technologien. Die Unternehmensarchitektur eines Unternehmens beschreibt diese Elemente und die vielfältigen Beziehungen zwischen diesen. In der heutigen Zeit, die von sehr dynamischen Märkten geprägt ist, stehen Unternehmen vor der Herausforderung schnell und agil auf verschiedene Markteinflüsse reagieren zu müssen. Dies betrifft insbesondere das Geschäftsmodell. Aufgrund der vielfältigen Beziehungen zwischen den Elementen einer Unternehmensarchitektur haben Änderungen des Geschäftsmodells zahlreiche Auswirkungen auf andere Elemente und Schichten der Unternehmensarchitektur. Das Enterprise Architecture Management (EAM) beschäftigt sich mit der Beherrschung und Weiterentwicklung der Unternehmensarchitektur [Ha13]. Hierbei liegt ein besonderes Augenmerk auf der Schaffung von Transparenz über die sehr komplexen Strukturen. Neben der organisatorischen Einbettung von EAM werden zur Unterstützung EAM-Werkzeuge genutzt, um die Unternehmensarchitektur zu dokumentieren und weiterzuentwickeln. Zur Aufdeckung von Optimierungspotentialen werden Analysetechniken eingesetzt [Ha13].

---

<sup>1</sup> Hochschule Reutlingen, {sebastian.breitbach, baris.cubukcuoglu, tobias.gorhan, daniel.hammer, christoph.wenzel }@student.reutlingen-university.de

<sup>2</sup> Hochschule Reutlingen, Herman Hollerith Zentrum, dierk.jugel@reutlingen-university.de

Das Analysieren von Unternehmensarchitekturen ist eine zeitaufwändige und auf vielen Informationen basierende Aufgabe. In der Praxis werden hierzu gewöhnlich Visualisierungen eingesetzt [Ha13]. Die EAM Werkzeugstudie von Matthes et al. [Ma08], sowie die EAM Visualisierungsstudie von Roth et al. [RZM14] zeigen, dass die von EAM Werkzeugen zur Verfügung gestellten Visualisierungen weitgehend statisch sind. Zur Verbesserung der Analysefähigkeit von Visualisierungen im EAM Umfeld haben Jugel et al. [JS14] den Ansatz eines interaktiven Architekturcockpits vorgestellt. Dieser Ansatz basiert auf Viewpoints, wie in [ISO11] beschrieben. Ein Viewpoint beschreibt nach [ISO11] die Konstruktion, Interpretation und Nutzung von Views. Views entsprechen dabei einer Visualisierung. Ein Beispiel dafür ist ein Clusterdiagramm. Die Besonderheit des Architekturcockpits ist die Möglichkeit eine Unternehmensarchitektur unter paralleler Nutzung zusammenhängender interaktiver Views analysieren zu können.

Wie lässt sich das in [JS14] beschriebene Konzept eines interaktiven Architekturcockpits in der Praxis umsetzen? Dazu stellen wir in diesem Artikel die Konzeption eines Prototyps vor der diese Frage beantworten könnte. Ein erster Schritt in Richtung eines Prototyps wurde in [Ki14] behandelt.

Durch ein Architekturcockpit ergeben sich vielseitige Möglichkeiten der interaktiven grafischen Analyse einer Unternehmensarchitektur. Dabei werden die verschiedenen Blickwinkel der an einem Entscheidungsprozess beteiligten Stakeholder berücksichtigt. Das Konzept eines Stakeholders ist ebenfalls im ISO Standard 42010 [ISO11] beschrieben. Ein Stakeholder entspricht nach dieser Definition einer Person oder Personengruppe, die einen oder mehrere sogenannter Concerns an der Unternehmensarchitektur hat. Ein Concern ist dabei ein Interesse, beispielsweise aufgrund einer Verantwortlichkeit oder besonderem Wissen über einen Ausschnitt der Unternehmensarchitektur. Die Visualisierung der Unternehmensarchitektur ermöglicht es komplexe IT-Unternehmensstrukturen verständlich und anschaulich abzubilden.

In [JS14] weisen Jugel und Schweda auf die Problematik der fokussierten beziehungsweise isolierten Sichtweise von Stakeholdern auf die Unternehmensarchitektur hin. Weiter heißt es, dass jeder Stakeholder Experte für seine Sichtweise ist und daher andere Anforderungen an die Betrachtung und Analyse einer Unternehmensarchitektur hat. Oftmals werden fachfremde Bereiche nicht berücksichtigt oder bedacht.

Wie eingangs erwähnt, können Änderungen innerhalb der Unternehmensarchitektur aufgrund der vielfältigen Beziehungen zwischen den Elementen weitreichende Auswirkungen auf andere Elemente haben. Aufgrund der hohen Komplexität einer Unternehmensarchitektur, sind diese Auswirkungen nicht ohne weiteres erkennbar, da sie nicht in direktem Kontext des betrachteten Architekturelements stehen. Dieser Umstand erschwert die Analyse von Unternehmensarchitekturen. Daher reicht das bloße Darstellen von Elementen nicht aus. Stattdessen müssen Beziehungen untereinander und mögliche Auswirkungen sichtbar gemacht werden. Entscheidungsrelevante Informationen sowie Einflüsse auf die Architektur müssen innerhalb der Visualisierung eindeutig zu identifizieren sein [JS14]. Wie in [Ju15b] aufgezeigt, fehlen in gängigen EAM Werkzeugen

Interaktionsmöglichkeiten bei Visualisierungen. Mithilfe eines interaktiven Cockpits soll dieses Problem gelöst und eine Interaktion mit der Unternehmensarchitektur ermöglicht werden. Dies soll ein tieferes Verständnis und eine bessere Entscheidungsfindung bei Änderungen der Architektur ermöglichen [Ju15a]. Die Kombination aus interaktiven Visualisierungen und automatisierten Analysen im Hintergrund wird als Visual Analytics bezeichnet [Ka08]. Dieser Ansatz findet bereits in vielen Anwendungsbereichen Anwendung. In der Domäne EAM fand dieser Ansatz bisher jedoch keine Anwendung. Unsere Arbeit konzentriert sich in dieser Iteration primär auf die interaktive Visualisierung der Unternehmens- und IT-Architektur, welche als Grundlage für weitere Arbeiten dient.

In Kapitel 2 beschreiben wir relevante Arbeiten zum Thema Visualisierung und Analyse von Unternehmensarchitekturen. Anschließend beschreiben wir in Kapitel 3 die Konzeption und in Kapitel 4 die Umsetzung des Prototyps. In Kapitel 5 soll Mithilfe eines Szenarios der Nutzen dargestellt werden.

## 2 Verwandte Ansätze

In diesem Kapitel werden verwandte Arbeiten mit Bezug zur Visualisierung von Unternehmensarchitekturen und zu visuellen Analysen vorgestellt.

Roth, Zec und Matthes [RZM14] betrachten in ihrer Visualisierungsstudie verschiedene Werkzeuge zur Visualisierung von Unternehmensarchitekturen, unter der Berücksichtigung unterschiedlicher Eigenschaften und Nutzungsmuster. Als Ergebnis der Studie kann zusammengefasst werden, dass Design, Konfigurationsmöglichkeiten sowie Anpassung des Informationsmodells wichtige Kriterien sind. Des Weiteren stellt sich heraus, dass die meist benötigten Visualisierungstypen Cluster Map, Matrix, Graph, Timeline und Flow-Diagramm sind. Weiter beschreibt die Studie das aktuelle Nutzerverhalten, zukünftige Anforderungen und bestehende Schwachstellen. Das Paper zeigt die wichtigsten Funktionen für EAM Werkzeuge mit Bezug zur Visualisierung auf. Diese Erkenntnisse bieten eine solide Grundlage innerhalb der Konzeption unseres Prototyps. Diese Arbeit lässt jedoch Anforderungen an die Visuelle Analyse von Unternehmensarchitekturen außer Acht. Das Prinzip von Visual Analytics wird in dieser Studie nicht erwähnt.

Schaub, Matthes und Roth [SMR12] stellen ein konzeptionelles Framework zur Entwicklung von interaktiven Visualisierungen vor. Hierbei steht die Manipulation der Unternehmensarchitekturdaten im Fokus. Aus diesem Grund werden u.a. Transaktions- und Konsistenzmechanismen benötigt. Da das Konzept des Architekturcockpits keine Manipulation des eigentlichen Modells der Unternehmensarchitektur vorsieht, sind solche Mechanismen nicht notwendig.

Naranjo, Sánchez und Villalobos [NSV13] beschreiben in ihrer Arbeit eine alternative Herangehensweise. Ihrer Ansicht nach sind einzelne Views ungeeignet um die gesamte

Unternehmensarchitektur zu beschreiben, da Views immer nur einen kleinen Teil der gesamten Architektur darstellen können. Ziel ist es einen Startpunkt zu finden um die Unternehmensarchitektur im Gesamtüberblick zu analysieren um mehr spezifische Analysetechniken zu entwickeln. Aufbauend auf [NSV13] beschreibt die Autoren in [NSV14] das PRIMROSe-Framework, einen konfigurierbaren und flexiblen Ansatz zur Analyse und Visualisierung von Unternehmensarchitekturen. Die Visualisierung und Analyse mit PRIMROSe besteht aus fünf Schritten: Import, Analyse, Map, Visualisierung und Kommunikation. Dabei ist jeder Schritt gleich aufgebaut: Input, Transformation und Output. Die Transformation wird dabei als Pipeline dargestellt. Mit Hilfe dieser Schritte und einem angepassten Metamodell, wird die Unternehmensarchitektur analysiert und visualisiert. In jeder Pipeline können entsprechende Analysemethoden formuliert werden, die unabhängig von benutzten Modellierungs-Tools, sequentiell ausgeführt werden können. Die Analyse beschränkt sich nicht auf die bereits existierenden Views sondern kann auf das gesamte Unternehmensmodell angewandt werden und präsentiert in den erstellten Views die Ergebnisse. Im Vergleich zu einem Architekturcockpit bietet dieser Ansatz jedoch nicht die Möglichkeit mehrere Views multiperspektivisch zu betrachten.

### 3 Konzeption

Unsere konzeptionellen Anforderungen basieren auf der Arbeit [JS14], in der bereits Funktionen für ein interaktives Cockpit beschrieben wurden. Eine wichtige Disziplin innerhalb der Planungsphase des Architekturcockpits ist die Erarbeitung einer adäquaten Architektur der Applikation. Ein Cockpit, auch „War Room“ oder „Situation Room“ genannt, war ursprünglich eine rein militärische Bezeichnung. Dort liefen alle relevanten Informationen zusammen um Staat und Streitkräfte effektiv, ohne Störungen von außen, führen zu können. Dieses Konzept wird heute auch in der Wirtschaft angewandt [Da06]. In unserem Fall sollen dabei alle relevanten Informationen über die Unternehmens- und IT-Architektur visualisiert und analysiert werden können. Die Darstellung der Informationen kann dabei auf mehreren Monitoren parallel erfolgen. Das Architekturcockpit ist in eine Frontend- und eine Backend-Applikation unterteilt. Das Frontend dient zur Visualisierung der Daten auf den Monitoren. Zwischen Backend und Frontend existiert ein Kommunikationskanal. Mithilfe dessen Hilfe werden JSON-Nachrichten ausgetauscht. Das nachfolgende Schaubild stellt die beschriebene Gesamtarchitektur dar:

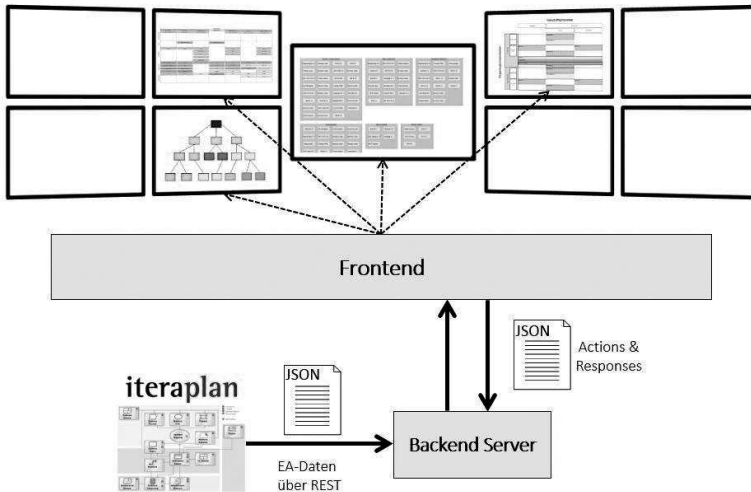


Abbildung 1:Konzeption des Architekturcockpits

In Hinblick auf die funktionalen und nichtfunktionalen Anforderungen, die in [Ju14] definiert sind, wurden die folgenden drei grundlegenden Komponenten des Frontends identifiziert: Datenmodell, Visualisierung und Kommunikation. Das nachfolgende Komponentendiagramm veranschaulicht den Aufbau des Frontends:

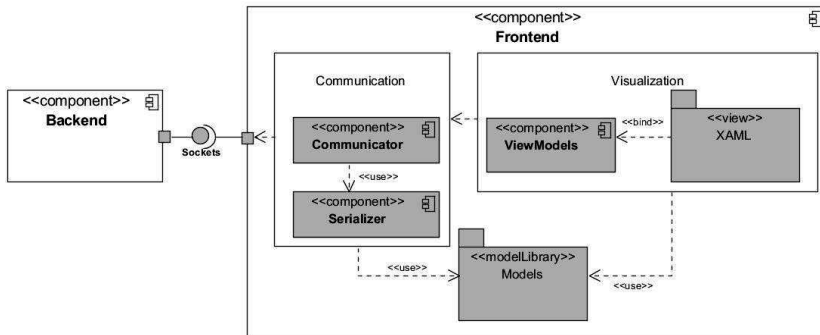


Abbildung 2: Komponentendiagramm des Frontends

Im Folgenden werden die einzelnen Komponenten detaillierter vorgestellt. Das *Datenmodell* muss im Rahmen einer Sitzung die Daten, welche vom Backend übermittelt werden, konsistent zwischenspeichern. Die Steuerung der Applikation sowie die Darstellung der Daten im Frontend werden innerhalb der *Visualisierungs*-Komponente zusammengefasst. Die Komponente *Kommunikation* fasst die Klassen zusammen, die zur synchronen und asynchronen Kommunikation zwischen Backend und Frontend benötigt werden.

Im Zuge der Erstellung eines ganzheitlichen Prototyps des Architekturcockpits wurde eine beispielhafte Unternehmensarchitektur im EAM Werkzeug iteraplan<sup>3</sup> abgebildet. iteraplan ist ein Werkzeug zur Verwaltung einer Unternehmensarchitektur, welches von der Firma iteratec GmbH entwickelt wurde. Das dort modellierte Modell einer Unternehmensarchitektur wird über eine vom Werkzeug bereitgestellte REST-Schnittstelle ins Backend geladen. Das Frontend greift auf diese Daten, die im Backend gespeichert werden, zu. Das Backend ist für die Aufbereitung des EA Modells und für die Erstellung von Views, die vom Frontend abgefordert werden, zuständig.

Das grundlegende Datenmodell zur Darstellung einer Unternehmensarchitektur bildet das sogenannte Atlas-Modell. Der Begriff des Atlas ist an den Atlas aus der Kartographie angelehnt. Dabei ist ein Atlas eine Sammlung verschiedener Blickwinkel auf die Erde. Jeder einzelne Blickwinkel ist dabei eine Visualisierung. Des Weiteren sind die Visualisierungen farblich codiert. Beispielsweise sind Wälder grün und Wasser blau. Ähnlich verhält es sich mit den Visualisierungen einer Unternehmensarchitektur. Mit Hilfe des in [JS14] beschriebenen Annotationsmechanismus können Architekturelemente annotiert und Annotationen visuelle Eigenschaften zugeordnet werden. Ein Atlas im Atlas-Modell bildet eine Unternehmensarchitektur ab, welche in iteraplan dokumentiert wurde. Eine Karte im Atlas-Modell, welches in [ISO11] als Viewpoint bezeichnet wird, erlaubt es die Architektur eines Unternehmens aus unterschiedlichen Blickwinkeln zu betrachten. Die Interessen der Stakeholder können durch variable Sichten adressiert werden. Diese Sichten werden als Views bezeichnet und sind bezüglich ihrer Granularität flexibel. Zur Visualisierung der Architektur werden unterschiedliche Strukturen verwendet. So können Hierarchien, Landscape- und Cluster-Diagramme zur Darstellung herangezogen werden. Innerhalb eines Views können eine Vielzahl von Ebenen, auch Layer genannt, angelegt werden. Die Layer werden benötigt, um das Problem von sich überlagernden visuellen Eigenschaften für ein bestimmtes Architekturelement zu lösen. Architekturelemente können ausgeblendet und farblich hervorgehoben werden. Ein Layer hat einen Namen, eine Beschreibung und ein oder mehrere Partitionen. Mit Hilfe der Partitionen können Architekturelemente innerhalb eines Layers gruppiert werden. Beispielsweise kann eine Partition erstellt werden, um Elemente einer bestimmten Annotation farblich hervorzuheben. In diesem Fall gibt es zwei Partitionen. Eine Partition für die Elemente der Annotation und eine für die restlichen Elemente. Beiden Partitionen können individuelle visuelle Eigenschaften zugeordnet werden. Darüber hinaus können Layer lokal auf einen View beschränkt werden oder global für alle Views im Architekturcockpit angewendet werden. So sind Beziehungen von Architekturelementen über verschiedene Views hinweg nachvollziehbar.

Das Frontend fordert verschiedene Views durch eine Benutzerinteraktion an. Die Views werden anschließend vom Backend erstellt und an das Frontend geliefert. Eine dauerhafte Speicherung dieser Daten im Frontend ist aufgrund der dadurch entstehenden Inkonsistenz nicht vorgesehen. Zur asynchronen und synchronen Kommunikation zwischen Back- und Frontend wird das sogenannte Action/Response-Prinzip eingesetzt. Dabei

---

<sup>3</sup> [www.iteraplan.de](http://www.iteraplan.de)

instanziiert eine Benutzerinteraktion eine Action. Der Inhalt einer Action ist der Name der initiierten Aktion, eine fortlaufende Nummer und Nutzdaten, welche für die Ausführung im Backend nötig sind. Die Action wird an das Backend gesendet. Anschließend erzeugt das Backend eine Response, durch den der aktualisierte Zustand des Atlas-Modells an das Frontend übermittelt wird. Dies kann entweder das vollständige Modell oder nur eine Teilmenge davon sein.

Für eine visuelle Planung der Unternehmensarchitektur, unter Einbeziehung unterschiedlicher Stakeholder, ist eine kollaborative Komponente unerlässlich. Dies setzt eine Mehrbenutzerfähigkeit und eine asynchrone Ereignissteuerung voraus. Für die Applikation wird das Publisher/Subscriber-Prinzip angewendet. Das Backend erfasst die eingeloggteten Teilnehmer, die am gleichen Atlas arbeiten. Hierbei spielt es keine Rolle, ob sich ein Benutzer im Cockpit anmeldet oder an einem normalen Computer. Der einzige Unterschied ist die Anzahl der Monitore. Bei einer Änderung, beispielsweise beim Hinzufügen eines Layers innerhalb eines Views, werden die Anzeigen der anderen Teilnehmer asynchron aktualisiert. Jeder Stakeholder arbeitet damit kontinuierlich auf dem aktuellen Zustand eines Atlas.

Zur Vorbereitung oder zur nachträglichen Änderung eines Atlas, hat das Frontend neben der Visualisierung auch eine Konfigurationsfunktion. Mit dieser lassen sich Views anlegen oder ändern. Wie die Darstellung der Unternehmensarchitektur, ist die Konfiguration ebenfalls geräteunabhängig und nach einer Speicherung sofort verfügbar.

## 4 Umsetzung

Als Grundlage für das Architekturcockpit wird das existierende Management Cockpit der Hochschule Reutlingen verwendet [Ki14]. Dieses besteht aus einem zentralen Touchscreen als Hauptbildschirm. Links und rechts von diesem befinden sich jeweils vier weitere Bildschirme.

Der Prototyp eines Frontends für das Architekturcockpit wurde mit C# und dem Framework WPF<sup>4</sup> entwickelt. Hierfür ist das .NET Framework<sup>5</sup> erforderlich. Die grafische Oberfläche basiert auf dem Pattern MVVM<sup>6</sup> und ermöglicht es das User Interface (View) unabhängig von der Programmlogik zu entwickeln. Die Daten, welche in der View angezeigt werden, werden mittels Data Binding in dieser platziert und so auf der Benutzeroberfläche dargestellt.

Die Applikation kann auf einem System mit einer beliebigen Anzahl an Monitoren ausgeführt werden. Die Steuerung dieser Applikation ist nur über den zentralen Hauptbildschirm möglich (siehe Abbildung 1). Eine Funktion verhindert, dass die Maus außerhalb

<sup>4</sup> Windows Presentation Foundation, Bibliotheken für GUI, <https://msdn.microsoft.com/de-de/library/ms754130%28v=vs.110%29.aspx>

<sup>5</sup> Laufzeitumgebung von Microsoft, stellt Klassenbibliotheken bereit, <http://www.microsoft.com/net>

<sup>6</sup> Model View ViewModel, <https://msdn.microsoft.com/en-us/library/gg405484%28v=pandp.40%29.aspx>

dieses zentralen Hauptbildschirms geschoben werden kann. Beim Starten der Applikation wird ermittelt wie viele Monitore dem Betriebssystem bekannt sind. Für jeden angeschlossenen Monitor wird ein Anwendungsfenster erzeugt und dieses auf dem Monitor platziert. Nach den Bildschirmkoordinaten werden die Anwendungsfenster sortiert und mit Indices in der Applikation bereitgestellt. Wird ein Atlas vom Benutzer geladen, werden die entsprechenden Views in den Anwendungsfenstern platziert. Die Bildschirmsteuerung wird in einem Overlay angezeigt und ermöglicht es, Views auf den seitlichen Bildschirmen zu verschieben, Views ohne Monitorzuordnung auf einen Monitor zu legen und Views auf den Hauptbildschirm in der Mitte zu übernehmen. Die Applikation kann über den berührungsempfindlichen Hauptbildschirm oder herkömmlich mit Tastatur und Maus gesteuert werden. Beim Einsatz einer Maus ist die Bedienung erleichtert, wenn diese nicht über viele Monitore geschoben werden muss und die zentrale Steuerung immer im Mittelpunkt stattfindet.

In der Kommunikationskomponente wird die Verbindung zum Backend über einen Socket aufgebaut. Dieser ermöglicht es, dass das Backend proaktiv Nachrichten an die verbundenen Clients senden kann. Im Frontend hört ein Listener, in einem separaten Thread, auf Push-Nachrichten des Backends bzw. auf Responses zu den gesendeten Actions. Dafür wurde bewusst ein separater Thread verwendet, so dass die GUI unabhängig vom Senden oder Empfangen ist. Das Backend sendet die Responses per Push-Nachricht an alle verbundenen Geräte für diesen Atlas.

Das Konzept der Actions und Responses basiert auf dem Pattern Request/Response. Das Architekturcockpit unterstützt 36 Actions. Beispiele für Actions sind das Einblenden von Zusatzinformationen zu einzelnen Elementen, das Annotieren von Elementen oder auch eine Impact Analyse auf ein einzelnes Element. Ein Klick auf ein Steuerelement löst eine Action aus, welche zum Backend gesendet wird. Hat das Backend eine Action verarbeitet, wird eine entsprechende Response mit aktualisierten Daten gesendet, welche im Frontend deserialisiert wird. Jede Klasse für eine Response beinhaltet eine Methode „Perform“, welche in der Kommunikationskomponente nach dem Eintreffen dieser Response ausgeführt wird und die Daten im Frontend aktualisiert. Die Views werden durch das Data-Binding beim Ändern der Daten aktualisiert.

## 5 Beispiel-Szenario

In diesem Kapitel wird ein fiktives Szenario beschrieben, um die Funktionen des Architekturcockpits exemplarisch darzustellen. Hierfür wird eine Sitzung im Architekturcockpit des fiktiven Unternehmens *Swabian Automotive AG* beschrieben. Diese ist ein deutscher Automobilhersteller für Luxusautos. An der Sitzung nehmen der CIO, der Unternehmensarchitekt, sowie der EA-Analyst teil. Der CIO stellte aufgrund eines ihm vorliegenden Berichts fest, dass die Kosten der IT sehr hoch sind und möchte Einsparungspotentiale identifizieren. Der Analyst beschreibt die aktuelle Situation der IT-Strukturen des Unternehmens und weist auf Einsparungspotentiale bei den Betriebssystemen hin.



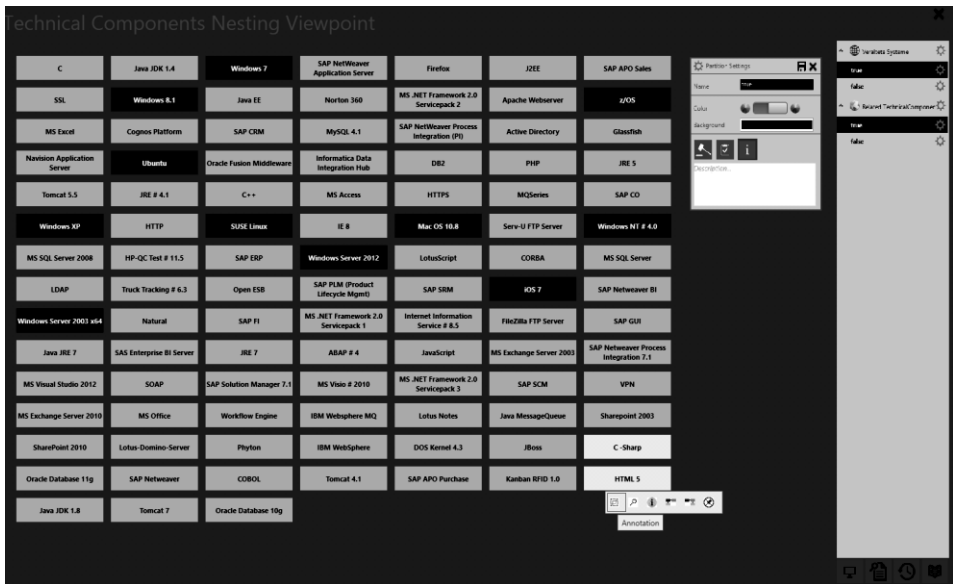


Abbildung 3: Hauptbildschirm: Darstellung der technischen Komponenten

Um dies zu untersuchen, blendet der Architekt den View „Architekturdomänen“ ein. Dieser wird als Cluster-Diagramm auf dem Hauptbildschirm des Cockpits positioniert. Das Diagramm zeigt alle Architekturdomänen und die jeweils zugehörigen technischen Bausteine. Um eine bessere Übersichtlichkeit der einzelnen Views zu erhalten, können diese an einem beliebigen Bildschirm des Cockpits positioniert werden. Das Verschieben und das Ein- und Ausblenden von Views ist ebenfalls möglich. Der Architekt berührt mit seinem Finger das Element „Operating Systems“ worauf ein schmales horizontales Menü mit Symbolen unterhalb des Elements erscheint. Er wählt das Symbol „Impact Analyse“. Daraufhin werden alle in Zusammenhang stehenden Elemente global auf allen Views eingefärbt und als Annotation in der Legende gespeichert. Die zu betrachtenden Betriebssysteme im View „Technical Components“ sind ebenfalls betroffen. Die Impact Analyse ermöglicht den Stakeholdern ein weitreichendes Verständnis über die gesamten Architekturelemente und deren Beziehungen zueinander. Der Analyst weist auf veraltete Betriebssysteme hin, die nur noch vereinzelt im Unternehmen genutzt werden, jedoch hohe Lizenzkosten verursachen. Der Architekt positioniert über die Navigation des Cockpits den View „Technical Components“ auf den Hauptbildschirm. Er markiert alle veralteten Betriebssysteme indem er sie jeweils mit dem Finger berührt. In dem Menü, das unterhalb des zuletzt markierten Elements erscheint, wählt er das Symbol Annotation aus. Daraufhin öffnet sich ein Overlay, in welchem er der Annotation die Bezeichnung „Veraltete Systeme“ vergibt und diese speichert. Der CIO bittet den Architekten der Annotation eine Notiz hinzuzufügen und diese als Information zu kennzeichnen. Der Architekt öffnet über die Legende die Einstellungen der Annotation „Veraltete

Systeme“. Hier gibt er die Notiz „Nutzer der veralteten Betriebssysteme über bevorstehende Abschaltung informieren“ ein und wählt das Symbol „Information“ an. Dieses Symbol wird daraufhin mit einer dunkleren Farbe hinterlegt, wodurch die Auswahl dargestellt wird. Der Architekt speichert die Änderungen, wodurch die Einstellungen automatisch ausgeblendet werden. Der CIO teilt dem Analysten die Aufgabe zu, die betroffenen Personen zu kontaktieren und beendet damit die Sitzung.

## 6 Fazit

Mit der Erstellung des Prototyps wurden Konzepte für die Darstellung und Interaktion mit einer Unternehmensarchitektur in einem Architekturcockpit erarbeitet und realisiert. Unter anderem können parallele Views auf mehreren Monitoren multiperspektivisch dargestellt werden. Das Cockpit kann sowohl per Maus als auch durch Berührungen des Hauptbildschirms gesteuert werden. Des Weiteren können Atlanten und Viewpoints durch die Nutzung eines Tablets oder direkt im Cockpit konfiguriert werden. Unterschiedliche Anwendungsszenarien, ähnlich dem beschriebenen aus Kapitel 5, haben den Prototyp evaluiert. Die Ergebnisse der Evaluierung zeigen, dass der Nutzen des Architekturcockpits die Erwartungen erfüllt und somit die Grundlage für weitere Projekte in diesem Bereich geschaffen wurde. Der Fokus dieser Arbeit lag auf der Visualisierung von Unternehmensarchitekturen mittels Views. Ein erster Schritt in Richtung Visual Analytics erfolgte durch die Implementierung der "Impact Analyse", die eine interaktive Visualisierung mit automatisierten Berechnungen im Hintergrund kombiniert.

Um den Prototyp zu erweitern, könnten in Folgearbeiten weitere Viewpoints implementiert werden. In Bezug auf die Interaktion mit den Visualisierungen und dem Cockpit selbst, sind weitere Funktionen denkbar. Beispielsweise könnte eine Show/Hide-Funktion eingebaut werden, mit der Elemente, zugunsten der Übersichtlichkeit, kurzfristig ausgeblendet werden könnten. Ebenso geplant ist eine Report-Funktion, welche eine Cockpit-Sitzung eigenständig dokumentiert. Diese könnte um weitere Funktionalitäten ergänzt werden. Durch die technischen Möglichkeiten, die das Cockpit bietet, ist eine Gesten- oder Sprachsteuerung ebenfalls vorstellbar. Außerdem sollte die Erweiterung und Absicherung der kollaborativen Nutzung des Cockpits in Betracht gezogen werden. Dafür wäre ein Rollenkonzept mit Berechtigungen von großem Nutzen. Weiter wäre eine Usability Study mit Stakeholdern aus realen Unternehmen wünschenswert um eventuelle Erfahrungswerte, der einzelnen Personengruppen, für die Verbesserung des Architekturcockpits nutzen zu können und die derzeitige Forschung an weiteren Analysemöglichkeiten an der Hochschule Reutlingen zu unterstützen.

## 7 Literaturverzeichnis

- [Da06] Daum, J.: Management Cockpit War Room: Objectives, Concept and Function, and Future Prospects of a (Still) Unusual, But Highly Effective Management Tool. In: Controlling – Zeitschrift für die erfolgsorientierte Unternehmensführung, vol. 18, C.H. Beck Verlag, München, Deutschland, 2006; Seiten 311-318
- [Ha13] Hanschke, I.: Strategisches Management der IT-Landschaft : Ein praktischer Leitfaden für das Enterprise Architecture Management. Hanser, München, Deutschland, 2013.
- [ISO11] ISO/IEC/IEEE 42010:2011 - Systems and software engineering - Architecture description; International Organization Of Standardization, 2011.
- [JS14] Jugel, D., Schweda, C.M.: Interactive functions of a Cockpit for Enterprise Architecture Planning. In: International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Ulm, Deutschland, 2014; Seiten 33-40
- [Ju15a] Jugel, D., Kehrer, S., Schweda, C., Zimmermann, A.: Towards Visual EAM Analytics – Automating Analyses in EAM. In: Digital Enterprise Computing 2015, 2015.
- [Ju15b] Jugel, D., Schweda, C., Zimmermann, A., Läufer, S.: Tool Capability in Visual EAM Analytics. In: Complex Systems Informatics and Modeling Quarterly (CSIMQ), 2015; Seiten 46-55
- [Ka08] Keim, D.A. et al.: Visual Analytics: Definition, Process, and Challenges. In: Information Visualization, Lecture Notes in Computer Science, vol. 4950, Springer Verlag, Heidelberg, Deutschland, 2008; Seiten 154- 175
- [Ki14] Kirchner, A., Scheurer, S., Weber, C., Wiechmann, A., Jugel, D.: Architektur eines Cockpits zur interaktiven Analyse von Enterprise Architectures auf Basis von Viewpoints. In: Hochschule Reutlingen Lecture Notes in Informatics, Gesellschaft für Informatik (GI), Volume S-13, Seiten 139 – 143, 2014.
- [Ma08] Matthes, F., Buckl, S., Leitel, J., Schweda, C.M.: Enterprise Architecture Management Tool Survey 2008, Technischer Bericht, 2008.
- [NSV13] Naranjo, D., Sanchez, M., Villalobos, J.: Connecting the Dots: Examining Visualization Techniques for Enterprise Architecture Model Analysis. In: International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Ulm, Deutschland, 2013; Seiten 29-38
- [NSV14] Naranjo, D., Sanchez, M., Villalobos, J.: Towards a Unified and Modular Approach for Visual Analysis of Enterprise Models. In: International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Ulm, Deutschland, 2014; Seiten 77-86
- [RZM14] Roth, S., Zec, M., Matthes, F.: Enterprise Architecture Visualization Tool Survey 2014. Technischer Bericht, Lehrstuhl für Informatik 19, Technische Universität München, 2014.
- [SMR12] Schaub, M.; Matthes, F.; Roth, S.: Towards a Conceptual Framework for Interactive Enterprise Architecture Management Visualizations. In: Modellierung, Bamberg, Deutschland, 2012.