

Entwicklung einer SAP Fiori Applikation am Beispiel eines “Schichteditor”

Dennis Fritsche
Professor Dr.-Ing. Frank Herrmann
Ostbayerische Technische Hochschule Regensburg
Innovationszentrum für Produktionslogistik und Fabrikplanung (IPF)
dennis.fritsche@st.oth-regensburg.de und frank.herrmann@oth-regensburg.de

Zusammenfassung—Ziel ist es die Entwicklung einer Webapplikation mithilfe des SAP User Interface for HTML 5 (SAPUI5) Frameworks anhand einer Beispielanwendung “Schichteditor” zu beschreiben. Diese Applikation soll die Pflege von Kapazitätsangeboten einzelner Maschinen für die Kapazitätsplanung im Rahmen der Produktionsplanung und -steuerung mithilfe des SAP Enterprise-Resource-Planning Systems vereinfachen und die SAP User Experience (UX) Strategie namens Fiori umsetzen. Außer der Zeitersparnis beim Pflegen der Kapazitätsdaten und der eingängigen Bedienung, soll die Applikation auf verschiedenen Endgeräten benutzt werden können und bietet dem Nutzer somit die Freiheit seine Arbeit fernab des klassischen Desktop PCs zu verrichten. Diese Anwendung wurde mithilfe der fachlichen Betreuung von T.CON realisiert.

I. MOTIVATION

Während Anwender im privaten Gebrauch vermehrt in ihrer Bedienbarkeit optimierte Anwendungen nutzen, welche neueste Erkenntnisse der Forschung an der Mensch-Maschinen Interaktion beachten, verwenden sie im arbeitstäglichen Umgang Software die fokussiert auf ihre Funktionalität entwickelt wird. Diese großen Unterschiede in der Software-Ergonomie führen zu veränderten Erwartungen und Anforderungen der Nutzer an Unternehmenssoftware. Ein Beispiel für diese auf Grund ihres Alters unzureichend zu bedienende Anwendung ist die Pflege von Schichtplänen zu einzelnen Arbeitsplätzen (Einstieg über die Anwendung “Arbeitsplatz ändern” Transaktion: CR02). Diese Anwendung kann nur mithilfe des SAP Graphical User Interface (SAP GUI) auf einem Desktop-PC gesteuert werden. Um den Schichtplan und somit auch die verfügbare Kapazität eines Arbeitsplatzes zu ändern, werden sehr viele Nutzeraktionen benötigt. Alleine die Auswahl der Schichten zu einem Arbeitsplan benötigt, wie in Abbildung 1 zu sehen ist, vier Aktionen - verteilt über drei Screens. Bei der Änderung von Schichtplänen zu mehreren Arbeitsplätzen summiert sich die Eingabezeit aufgrund der fehlenden Funktion einer Massenbearbeitung. Die Schichtänderungen selbst lassen sich nur komplex und nicht intuitiv bedienen. Wie in Abbildung 2 verdeutlicht wird muss der Nutzer anhand seines Datums überlegen in welchem Datum-Intervall der Schichtplan hinzugefügt werden muss. Für die Änderung muss ein eigenes Datumsintervall angelegt werden. Anhand einer Wochenstruktur muss der Nutzer nun herausfinden, welches

Datum welchem Wochentag entspricht. Möchte er beispielsweise am 01.06.2017 eine Frühschicht hinzufügen, so muss er in einem Kalender nachsehen, dass es sich um einen Donnerstag handelt, um die gewünschten Änderungen vornehmen zu können. Dem Nutzer liegen also nicht alle Informationen vor um seine Aufgabe effektiv bearbeiten zu können. Diese Änderung benötigt der Nutzer ebenfalls mindestens vier Aktionen. Sofern der Nutzer Schichten nach Kapazitätsbedarf ändern möchte, existiert keine Anzeige, inwieweit der Kapazitätsbedarf durch das Angebot gedeckt werden kann. Der Nutzer muss also zwischen der Kapazitätspflege-Anwendung und der “Erweiterte Kapazitätsauswertung: Arbeitsplatzsicht” (Transaktion: CM50) wechseln, um die Auswirkungen seiner Änderungen nachvollziehen zu können (siehe Abbildung 3).

II. ZIELSETZUNG

Die Applikation wird nach der Nutzungserlebnis-Strategie Fiori von SAP entwickelt. Durch diese Strategie wird für alle Anwendungen innerhalb der SAP Business Suite das gleiche “Look and Feel” erreicht. Durch ein verbessertes User Interface (UI) spart der Anwender Zeit beim Erlernen neuer Anwendungen und seine Produktivität wird gesteigert. Das neue UI hat nach (Bince, 2015) folgende Vorteile gegenüber der altbewährten Benutzeroberfläche:

- Weniger Nutzungsfehler
- Geringere Kosten durch Anwenderschulungen
- Höhere Zufriedenheit der Nutzer

Die zu entwickelnde Anwendung soll einen schnellen Überblick über die Schichtzeiten bieten und einfach bedienbar sein. Dem Nutzer soll klar sein, welche Auswirkungen seine Änderungen haben. Der Nutzer kann schnell zwischen den zu bearbeitenden Arbeitsplätzen wechseln. Durch Kontrollen wird sichergestellt, dass der Nutzer keine falschen Eingaben tätigen kann. Die Anwendung soll ohne größere Modifikationen an verschiedene Kunden verteilt werden können. Durch die Nutzung des SAPUI5 Frameworks wird die Bedienung der Fiori Applikation “Schichteditor” zusätzlich über ein Tablet ermöglicht. Bei der Applikation handelt es sich also um eine Webapplikation die einen REST-Service (der ebenfalls entwickelt wurde) konsumiert. Für die Verständlichkeit wird

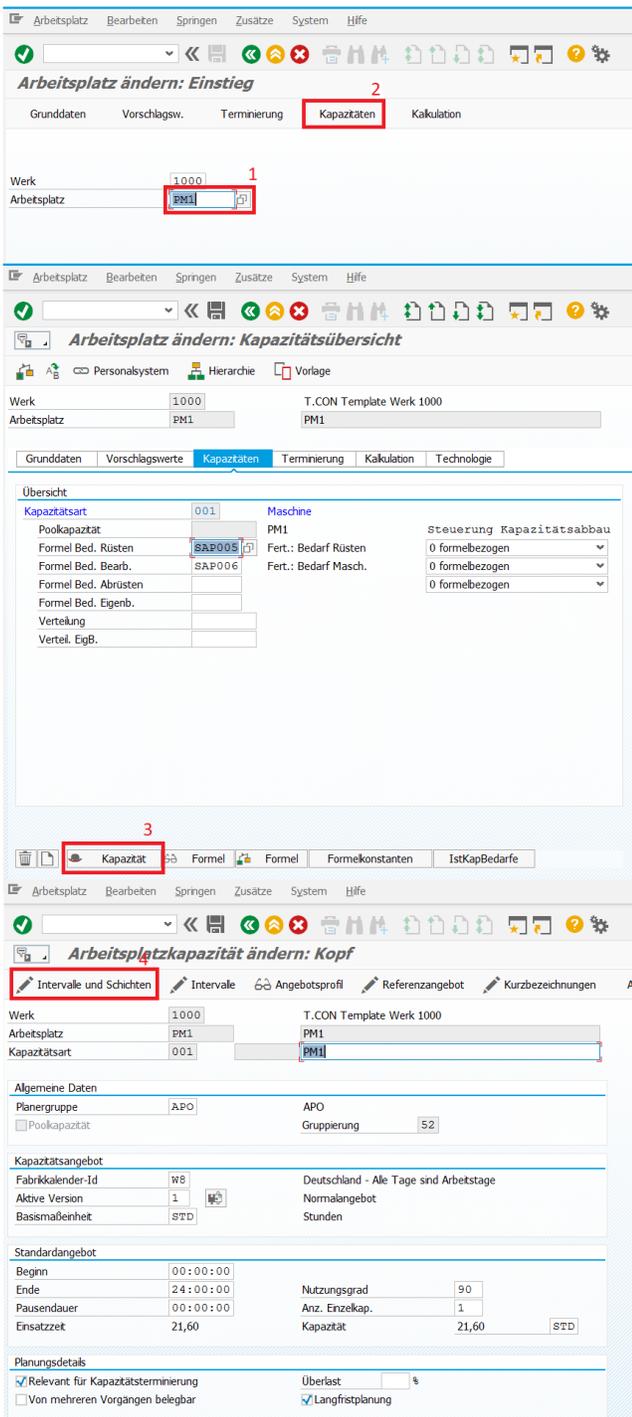


Abbildung 1. Vollständige Navigation zu den Schichten eines Arbeitsplatzes über 3 Screens

zuerst auf die Grundlagen einer Fiori Applikation eingegangen. Dies beantwortet die Frage, wann eine Applikation die SAP UX Strategie Fiori umsetzt. Anschließend werden kurz die Komponenten einer Fiori Systemlandschaft und ihr Einfluss auf den Entwicklungsprozess vorgestellt. Auch eine kurze Beschreibung der Softwarekomponente SAP Netweaver Gateway erfolgt. Kurz wird auf das genutzte Framework

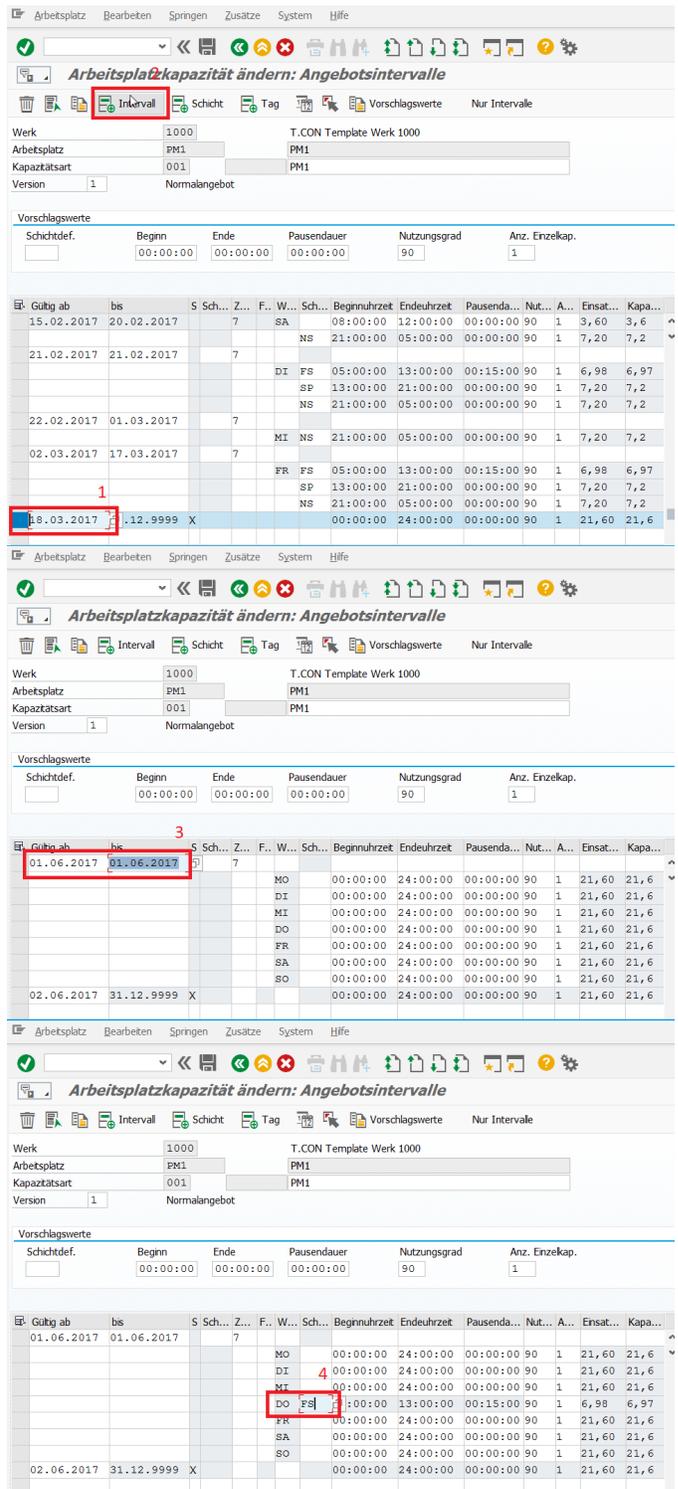


Abbildung 2. Hinzufügen einer Schicht am 01.06.2017

und den Betriebswissenschaftliche Umfeld der Applikation eingegangen. Im letzte Teil wird die Entwicklung der Applikation und des Service beschrieben, dazu müssen zuerst die Oberfläche entworfen und die benötigten Stammdaten und Bewegungsdaten identifiziert werden. Abschließend zur

Einträge	KapBed. Rest	KapAngebot	KBel.	Fr. Kapaz. Rest
Summe	100,00 %	100,00 %		100,00 %
31.05.2017	0,00 %	0,78 %	0,0	0,78 %
01.06.2017	0,00 %	0,78 %	0,0	0,78 %
02.06.2017	0,00 %	0,78 %	0,0	0,78 %
03.06.2017	0,00 %	0,78 %	0,0	0,78 %
04.06.2017	0,00 %	0,78 %	0,0	0,78 %
05.06.2017	0,00 %	0,78 %	0,0	0,78 %
06.06.2017	0,00 %	0,78 %	0,0	0,78 %
07.06.2017	0,00 %	0,78 %	0,0	0,78 %

Abbildung 3. Darstellung der Kapazitätsauslastung zu einem Arbeitsplatz

Beschreibung der Konzepte und Implementierung von Service und Applikation erfolgt eine Bewertung des Ergebnis.

III. GRUNDLAGEN

Nachfolgend wird auf die Grundlagen für die Entwicklung einer SAP Fiori Applikation und deren Einfluss auf den Entwicklungsprozess des "Schichteditor" eingegangen. Dazu müssen die Grundsätze der Fiori UX, der Aufbau einer Fiori Systemlandschaft, die Funktion des SAP Netweaver Gateway und die Grundsätzlichen Konzepte der Entwicklung mit dem SAPUI5 Framework erläutert werden. Zusätzlich wird das Betriebswirtschaftliche Umfeld des "Schichteditors" erklärt.

A. Fiori Grundsätze

Für die Fiori UX Strategie wurden von SAP fünf Grundsätze festgelegt, welche eine Applikation erfüllen muss, um Fiori konform zu sein:

- Rollenbasiert: Jede App bedient genau die Rolle des Benutzers. Es werden ihm nur die Funktionen angeboten, welche er für die Durchführung seiner Aufgabe in seiner Rolle benötigt und er erhält die Informationen die er benötigt zur passenden Zeit.
- Responsive Design: Die Anwendung kann in ihrer Anzeige und Funktion auf verschiedenen Ausgabegeräte reagieren und akzeptiert dementsprechend verschiedene Eingabemöglichkeiten. Die Anwendung ist mit verschiedenen Bildschirmgrößen gut bedienbar und das Layout passt sich automatisch an. Die Anwendung funktioniert auf den verschiedensten Plattformen gleich.
- Einfach: Anwendungen bieten keine Funktionen, die nicht zwingend für die Bearbeitung der Aufgabe nötig sind. Innerhalb von maximal drei Bildschirmbildern soll der Anwender einen Anwendungsfall bearbeiten können.
- Kohärent/Stimmig: Der Anwender findet sich aufgrund von Design und Aufbau schnell zwischen verschiedenen Fiori Anwendungen zurecht. Er kann bereits Gelerntes

auf neue Anwendungen anwenden und erkennt Muster, die ihm helfen neue Aufgaben intuitiv zu erledigen.

- Ansprechend: Anwendungen werden nach Designvorgaben erstellt und sind optisch ansprechend. Dafür werden Fiori Design Guidelines genutzt. Diese legen fest, welche UI Design Pattern für Anzeige und Bedienung genutzt werden sollen.

B. Komponenten der SAP Fiori Systemlandschaft

Für die Entwicklung einer Fiori Applikation ist Wissen über die genutzte Architektur nötig. Dadurch wird verständlich in welchen Komponenten sich die Softwareartefakte befinden und wie diese zusammenarbeiten. Dies ist nicht nur wichtig für das Verständnis der Implementierung (und dem Testen), sondern für das Verständnis der Architektur der Applikation. Die klassische SAP Fiori Systemlandschaft wie in Abbildung 4 zu sehen aus bis zu sechs verschiedenen Schichten:

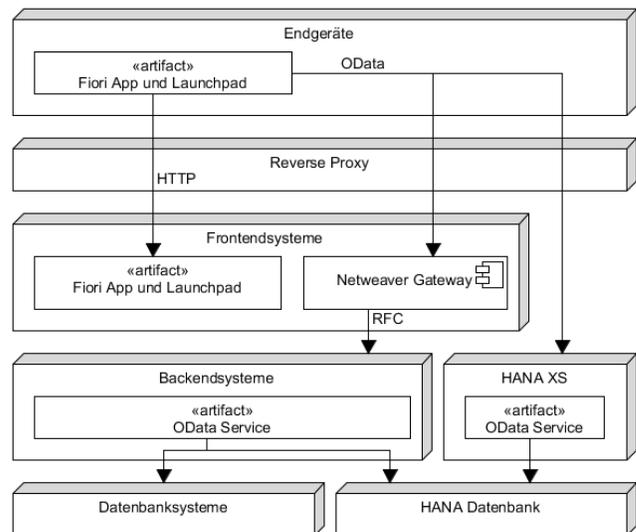


Abbildung 4. Vollständig ausgeprägte Fiori Landschaft mit den wichtigsten Artefakten, Komponenten, Protokollen und Zugriffsrichtungen

- Endgeräte, welche entweder mit dem Browser oder speziellen Apps auf die SAP Fiori Applikationen und über einen REST-Service auf die Geschäftsdaten zugreifen.
- Frontendsystem, welches die Ressourcen für das Fiori Launchpad, dem Homescreen für Fiori Apps als zentraler Einstiegspunkt für den "Schichteditor", und die Fiori Applikationen selbst zur Verfügung stellt. Des Weiteren dient es als Gateway für den Zugriff auf die Geschäftslogik und Daten aus den Backendsystemen.
- Backendsystem auf dem SAP ERP System installiert ist.

Als Zugriffsclients für die Applikation wurden für die Entwicklung und den produktiven Einsatz der Browser von Desktop und Tablet festgelegt. Auf Grund der Menge der anzuzeigenden Daten und ihrer Darstellungsweise ist eine

Smartphone Anwendung im praktikabel, da die Bildschirmfläche dafür zu klein ist und dem Ziel der Anwendung einen schnellen Überblick über die Schichtzeiten zu bieten nicht entsprechen kann. Wie in Abbildung 5 dargestellt gibt es neben

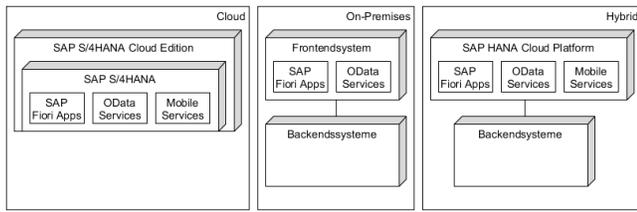


Abbildung 5. Gegenüberstellung Fiori Landschaft anhand des Lizenzierungsmodells

einer klaren On-Premises Lösung und einer Cloud Lösung ein hybrides Modell. In einem Hybrid System werden die Frontendsysteme mithilfe der SAP HANA Cloud Platform (SAP HCP), dem Zugriffsportal auf SAP Cloud Dienste, verwaltet. Backendsysteme und Datenbanken bleiben unter der Verwaltung des Lizenznehmers. Die Fiori Architektur des Entwicklungssystems für die "Schichteditor" Applikation ist hybrid (siehe Abbildung 5). Frontendserver und Web Dispatcher werden also innerhalb der SAP Cloud gehostet, ein Backendsystem in Verwaltung von TCONs wird als Quelle für Testdaten und für die Service Entwicklung genutzt. Das Fiori Launchpad wird ebenfalls über die SAP Cloud gehostet. Als Servicebereitstellungsmodell wird wie also "Gateway as a

Service" genutzt (siehe Abbildung 6). Dafür muss der Cloud Connector genutzt werden. Der Cloud Connector baut einen über SSL gesicherten Tunnel des Backendsystems zu der SAP HCP auf. Für eine bessere Verteilung der App würde es sich auch anbieten den Service zu "Central Hub Deployment - Development on the Hub" zu modifizieren. Dadurch würde der Service mobiler und könnte in allen möglichen Fiori Landschaftsausprägungen eingesetzt werden. Dies wurde in der Entwicklung des Service noch nicht beachtet, kann aber nachgebessert werden. Dazu müsste alle direkten Lesezugriffe auf die Datenbank mithilfe von RFC-Bausteinen erledigt werden. Die Serviceimplementierung der "Schichteditor" Applikation befinden sich also auf dem Backendsystem, der Service selbst, wird über HCP bereitgestellt. Über das HCP wird ebenfalls die Anwendung und das Fiori-Launchpad als Einstiegspunkt zu Verfügung gestellt. Durch diesen Aufbau und der Abgrenzung zwischen Service und Servicekonsumenten ist die Fiori Applikation sowohl als Desktop, als auch Tablet Anwendung verfügbar und kann leicht ohne größere Anpassung an verschiedene Kunden verteilt werden.

C. SAP Netweaver Gateway

Bei SAP Netweaver Gateway handelt es sich um eine technologische Schnittstelle zu der SAP Netweaver Plattform, genauer um eine REST Schnittstelle für SAP Netweaver Application Server. Sie dient als zentrale Komponente eines SAP Systems für Angebot und Konsum von Webservices. Bei dem Entwicklungssystem für den Service handelt es sich um einen Netweaver Application Server. Das Open Data Protocol wird als Protokoll für die Bereitstellung und Nutzung des REST Service für die "Schichteditor" Applikation genutzt.

D. Konzepte SAPUI5

SAPUI5 ist ein Clientseitiges UI Framework für die Entwicklung Betriebssystemunabhängiger Geschäftsanwendungen. Es basiert auf HTML5, CSS3 und JavaScript. Die Anwendungen werden Clientseitig mithilfe von Javascript ausgeführt und sind somit unabhängig vom Server. Das Framework ist "Controls"-basiert und nutzt als Designgrundlage das MVC-Model. Zusätzlich wird das "Hypermedia As The Engine Of Application State"-Prinzip (HATEOAS) unterstützt und kann leicht internationalisiert und das Aussehen über Themes angepasst werden.

E. Betriebswirtschaftliches Umfeld

Im Rahmen der Fertigungsauftragseröffnung im SAP ERP System wird bei der Berechnung der Ecktermine des Fertigungsauftrags der Kapazitätsbedarf des Auftrags an den Produktionsleistungen der fertigenden Maschinen bestimmt. Dazu muss die Menge des zu produzierenden Materials und der terminierte Arbeitsplan für die Erstellung dieser Menge des Materials mit den Durchführungszeiten der

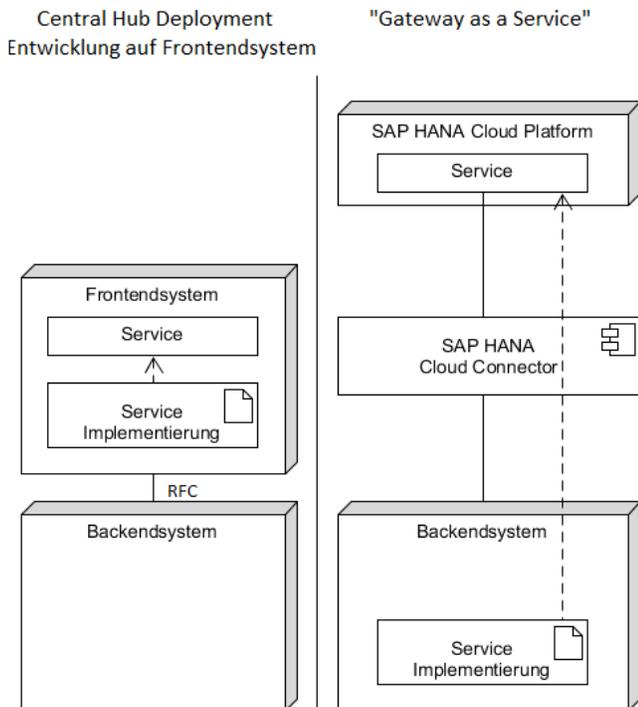


Abbildung 6. Servicebereitstellungsmöglichkeiten innerhalb einer Fiori Systemlandschaft

nötigen Arbeitsvorgänge an der durch den Arbeitsplan bestimmten Maschinen verrechnet werden. Der Kapazitätsbedarf kann über eine längere Zeitperiode bestehen, ein Verteilungsschlüssel bestimmt wie sich der Bedarf über den Zeitraum verteilt. Die Kapazitätsauslastung der einzelnen Arbeitsplätze kann abhängig von Auftragsart und Kapazitätsart sowohl tabellarisch als auch graphisch mithilfe von verschiedenen Transaktionen ausgewertet werden. Durch die Kapazitätsverfügbarkeitsprüfung, welche bei einer Auftragseröffnung, -änderung oder -freigabe durchgeführt werden kann, wird festgestellt ob im gewünschten Zeitraum ausreichend Kapazitäten existieren um den Kapazitätsbedarfe zu decken. Falls nicht genügend Kapazität vorhanden ist, kann mithilfe der Kapazitätsterminierung ein Alternativtermin bestimmt werden.

IV. ENTWURF GRAPHISCHE OBERFLÄCHE

Bei dem Entwurf der Oberfläche der Anwendung werden die SAP Fiori Design Guidelines eingehalten und vorgeschlagene Floorplans und Designpatterns genutzt. In Abbildung 7 ist diese Skizze der Anwendung dargestellt. Die App soll einen schnellen Überblick auf die Schichtdaten der einzelnen Kapazitäten zu den Arbeitsplätzen bieten. Dazu wird als erstes ein Arbeitsplatz ausgewählt (siehe (1) in Skizze). Es wird der Arbeitsplatz mit Detailinformationen und allen verfügbaren Kapazitäten angezeigt (2). Die einzelnen Kapazitäten werden durch Tabs verbildlicht (3). Bei der Auswahl einer Kapazität (durch Tab Auswahl) werden die Schichtdaten für einen bestimmten Zeitraum mithilfe einer Tabelle angezeigt (4). Jede Zeile in der Tabelle entspricht einen Tag, jede Spalte entspricht einer möglichen Schicht. Der Nutzer kann nun mithilfe von Checkboxes auswählen, ob Schichten stattfinden. Den Rahmen der Applikation bildet die Fiori Launchpad Shell Bar (5). Als grundlegenden Floorplan wurde das Split-Screen Layout gewählt, da der Nutzer schnell zwischen verschiedenen Arbeitsplätzen wechseln können soll. Eine komplexe Filterung der Arbeitsplätze ist nicht nötig, deswegen können alle verfügbaren Arbeitsplätze als Liste angeboten werden. Durch dieses Layout wird die Oberfläche in eine „Master list area“ für die Liste der Arbeitsplätze und eine „Details area“, für die Anzeige der Schichtwerte aufgeteilt. Am oberen Rand der der „Master list area“ befindet sich eine Suchleiste für die Suche innerhalb der Arbeitsplatzliste, am unteren Rand wird eine Toolbar mit Funktionen für die Gruppierung und Reihenfolge dieser Liste angeboten. In der „Detail area“ wird der „Object Page“ Floorplan angewendet. Dieser Floorplan ist dafür geeignet einfache Objekte, ohne komplexer Navigation, anzuzeigen und diese für die Bearbeitung anzubieten. In den oberen Teil dieses Bereichs befindet sich eine Übersicht zu welchem Arbeitsplatz gerade Änderungen vorgenommen werden. Sobald eine Kapazität ausgewählt ist, wird die Tabelle mit den Schichtwerten angezeigt. Im unteren Bereich bietet eine Toolbar Bearbeitungsfunktionen zu der Schichttabelle an. In der Skizze ist noch kein graphischer Indikator für die Auswirkung der Schichtänderung auf die Kapazitätsauslastung

vorhanden. Diese Anzeige wurde in einer späteren Iteration im Entwurf hinzugefügt.

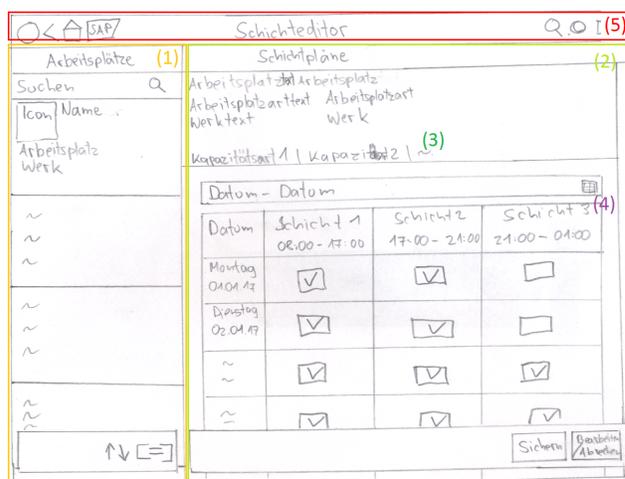


Abbildung 7. Skizze „Schichteditor“

V. STAMMDATEN

Durch die Skizze wurde eine Reihe von Stammdaten identifiziert, welche für die Applikation benötigt werden. Die für das Verständnis der Applikation und für die Entwicklung am wichtigsten sind Arbeitsplatz, Kapazität und Schichtgruppe.

- Arbeitsplätze sind Struktureinheiten eines Unternehmens in denen Arbeitsaufgaben erfüllt werden. In SAP ERP bildet der Arbeitsplatz nach (Dickersbach & Keller, 2010) je nach betrieblicher Organisation Unternehmensressourcen wie Maschinen, Gruppe von Maschinen, Personen oder Gruppen von Personen. Arbeitsplätze werden für die Kalkulation von Fertigungskosten, für die Durchlaufterminierung und für die Kapazitätsplanung verwendet. Vorgänge aus Aufträge werden an Arbeitsplätzen ausgeführt. Durch einen Arbeitsplan findet diese Zuordnung statt. Der ERP-Arbeitsplatz beschreibt also den örtlichen Raum an dem die Arbeit stattfindet. Arbeitsplätze werden einem Werk zugeordnet (siehe Abbildung 8). Dieses ist seine Betriebsstätte. Die Arbeitsplatzart beschreibt in welchen Plänen der Arbeitsplatz genutzt werden kann.
- Die Kapazität ist in SAP ERP das quantitative Leistungsvermögen eines einzelnen Arbeitsplatzes, oder mit einer Poolkapazität, das von mehreren Arbeitsplätzen. Kapazitäten besitzen eine Kapazitätsart welche genauer beschreibt, um welche Kapazität es sich handelt. Neben Maschinenkapazität und Personalkapazität lassen sich hiermit beispielsweise auch Reservekapazitäten oder aber auch Kapazitäten des Energieverbrauchs modellieren. Um das Kapazitätsangebot zu berechnen, werden die Einsatzzeiten hinterlegt. Diese legen fest wann diese Kapazität einsatzfähig ist. Kapazitätsangebote werden auf zwei verschiedenen Arten gepflegt, entweder

mithilfe des Standardangebots, welches für jeden Arbeitstag der Woche gleich eine Einsatzzeit definiert, oder feingranularer, mithilfe von Schichten. Dafür werden mit Datumswerten Intervalle gebildet. Diese legen fest, wann welche Einsatzzeitdefinitionen gelten. Für die Kapazitätsbedarfserrechnung und die Terminierung der einzelnen Vorgänge auf einem Arbeitsplatz werden in der Kapazität Formeln für deren Errechnung hinterlegt.

- Schichtgruppe Mithilfe der Schichtgruppe werden die einzelnen Definitionen zur genauen Bestimmung der Schichtzeiten zusammengefasst. Dazu gehören das Schichtprogramm, die Schicht- und die Pausenzeiten. Bei einem Schichtprogramm wird festgelegt, welche Schichten an den unterschiedlichen Wochentagen stattfinden. Die Schichtdefinitionen bestimmen den Zeitraum der Schichten und die Pausenzeiten die Pausen. Die Schichtgruppe wird Kapazitäten zugeordnet. Diese haben somit, bei der Pflege der Angebotsintervalle, Zugriff auf die Schichtdefinitionen. Mit den Gruppierungen wird somit eine schnellere Pflege von Kapazitätsangeboten einer Kapazität ermöglicht.

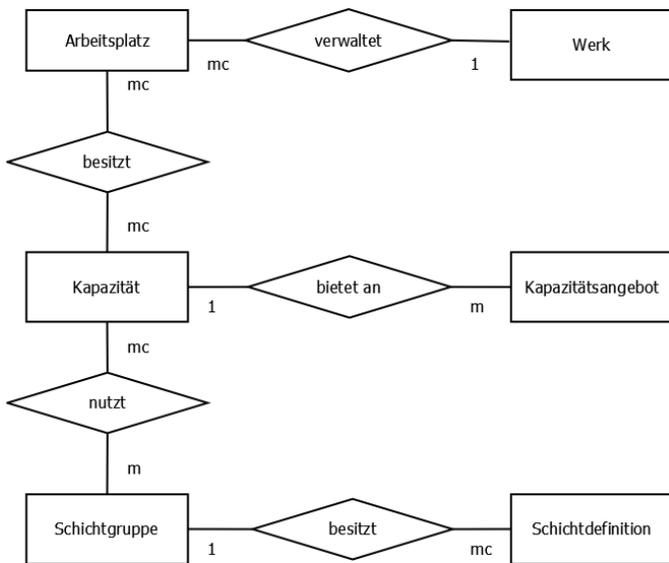


Abbildung 8. Entity-Relationship-Modell der Stammdaten mit den wichtigsten Abhängigkeiten

VI. SERVICE

A. Konzept

Die Grundsätzliche Idee der Anwendung ist es, den Großteil der Berechnungen auf dem Backendsystem (Programmierung in ABAP) durchzuführen. Die Frontend Anwendung (Programmierung in Javascript) ist nur zuständig für die Präsentation der Daten und für die Verarbeitung der Benutzerinteraktionen. Abbildung 9 verbildlicht die Abhängigkeiten zwischen den verschiedenen im Service

angebotenen Entitätstypen. Nicht alle Relationen werden implementiert (OData-Grundsatz: “Der Service muss nicht alle möglichen Funktionen implementieren um in seiner Basis Funktion nutzbar zu sein”). Es existiert eine Service

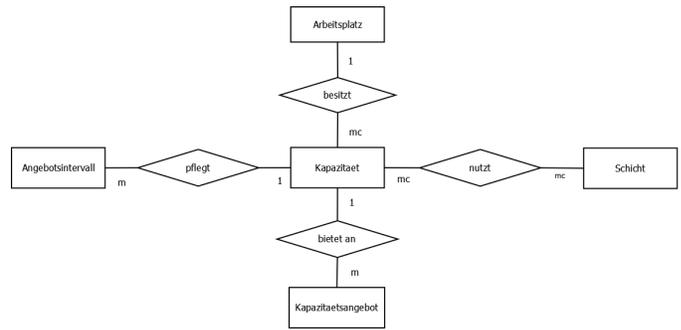


Abbildung 9. Entity-Relationship-Modell des Services

Adresse für die Auflistung aller Arbeitsplätze die mit dem Backendsystem verwaltet werden (Service URL + /ArbeitsplatzSet). Diese dient als Einsprungadresse in die Anwendung (HATEOAS). Es können aber auch einzelne Arbeitsplatzentitäten direkt abgerufen werden. Für die Eingrenzung und Sortierung der Arbeitsplätze wird eine Filter- und Sortierfunktion implementiert. Die Filterfunktionen ermöglicht auch die Suche innerhalb der Arbeitsplätze nach Werk und Arbeitsplatz. Zusätzlich zur Sortierung kann auch die Gruppierung der Liste vorgenommen werden. Mithilfe einer “expand” Funktion können zu einem Arbeitsplatz alle über eine Fremdschlüsselbeziehung zugeordneten Kapazitäten ausgegeben werden. Zu den Kapazitäten (im Service “Kapazitaet” genannt) wird nur über die Fremdschlüsselbeziehung navigiert. Die Adresse der Entitätsmenge kann nicht zum Einstieg in den Service verwendet werden. Dies gilt auch für die restlichen Entitätstypen. Es müssen aber trotzdem der Abruf aller und einzelner Kapazitätsentitäten ermöglicht werden, damit sie über die Navigation mit der Fremdschlüsselbeziehung trotzdem abrufbar sind. Um die Schichtdefinitionen (im Service “Schicht” genannt) zu einer Kapazität zu erhalten wird die Gruppierung der Kapazität benötigt. Da aber nur interessant ist, welche Schichtdefinitionen in dieser Gruppe enthalten sind, wird sie von dem Service nicht angeboten. Schichtdefinitionen besitzen einen Zeitraum, in welchem sie gültig sind. Die Benutzerauswahl des abzurufenden Zeitraums der Schichtwerte (oder standardmäßig drei Wochen ab Tag der Nutzung) definiert eine Filterfunktion welche die für den Zeitraum gültigen Schichtdefinitionen abfragt. Eine Kapazität besitzt verschiedenen Versionen des Kapazitätsangebots (im Service “Kapazitaetsangebot” genannt), über die Kapazität wird definiert welche Version des Kapazitätsangebots gerade für die Planung aktiv ist. Mit der Anwendung wird nur das aktive Kapazitätsangebot bearbeitet. Kapazitätsangebote haben ebenfalls einen Zeitraum, in denen sie gelten. Der Abruf aller gültigen Kapazitätsangebote erfolgt mit einer Filterfunktion. Für die Bearbeitung dieser Werte (Auswahl und Abwahl bestimmter Schichten führt zu einer Veränderung das Kapazitätsangebots)

sollte die HTTP Update Methode genutzt werden. Für die Änderung des Kapazitätsangebots müssen aber zugleich die Angebotsintervalle verändert werden. Angebotsintervalle sind für die Nutzung der Schichtpflegeanwendung im SAP ERP System nötig. Für die Verbuchung des Kapazitätsangebotes wird ein Funktionsbaustein SAPs aufgerufen, welcher die neuen/veränderten Werte prüft und anschließend schreibt. Dieser Funktionsbaustein benötigt zur gleichen Zeit die zu ändernden Kapazitätsangebote und Angebotsintervalle. Da aber erst zur Laufzeit der Anwendung genau bekannt ist, wie groß der maximale Zeitraum der zu bearbeiteten Kapazitätsangebote und somit auch der Angebotsintervalls ist, ist nicht bekannt welche Angebotsintervalle verändert und welche gelöscht werden müssen. Deswegen wird von der Schichteditor Anwendung eine Kombination aus löschen und erstellen genutzt. Der komplette Zeitraum, in dem Änderungen stattfinden wird gelöscht, Clientseitig werden alle Werten, die sich in einem Intervall mit dem kleinsten und größten Datum mit geänderten Schichtwerten befinden zusammengefasst (innerhalb einer Wochenstruktur, Abbildung 10 verdeutlicht dieses Prinzip) und als neues Intervall erstellt. Das löschen erfolgt mithilfe eines "Function Import", das erstellen der Kapazitätsangebote und Angebotsintervalle mithilfe von einer HTTP Create Methode.

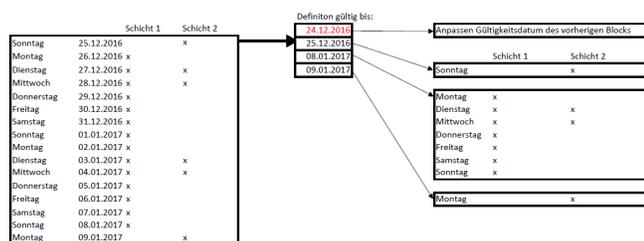


Abbildung 10. Beispiel für die Komprimierung des Kapazitätsangebots für eine Änderung von stattfindenden Schichten zwischen 24.12.2016 bis 09.01.2017

B. Implementierung

Für die Serviceimplementierung wird zuerst der Service und das benutzte Datenmodell definiert. Nach der Definition des Datenmodells können Laufzeitartefakte generiert werden. In Tabelle 2 ist eine Auflistung der generierten Artefakte eines Services und ihre Namens Kürzel dargestellt. Die Model-Provider-Klassen repräsentieren das Datenmodell zur Laufzeit. Für eine einfache Erweiterung der Funktionalität über Redefinitionen existiert eine Aufteilung in Basis- und Erweiterungsklasse. In den Data-Provider-Klassen findet die tatsächliche Implementierung der gewünschten Funktionen des Service statt. Hier existiert ebenfalls eine Aufteilung der Klassen für die einfache Redefinition. Wenn der Service im Gateway System registriert wird, und somit zum Aufruf bereitgestellt wird, werden noch zwei weitere Artefakte für den registrierten Service und das registrierte Modell erstellt.

Tabelle 1

GENERIERTE ARTEFAKTE FÜR DIE LAUFZEIT EINES SERVICE

Artefakttyp	Kürzel
Model-Provider-Basisklasse	MPC
Model-Provider-Erweiterungsklasse	MPC_EXT
Data-Provider-Basisklasse	DPC
Data-Provider-Erweiterungsklasse	DPC_EXT

- `/IWBEP/IF_MGW_APPL_SRV_RUNTIME~EXECUTE_ACTION` Implementierung des Function Import "delete_kapa_kazy" für das gleichzeitige löschen von Angebotsintervall und Kapazitätsangebot. Die durch das Datum in den Parameter des Function Import angegebenen Werte grenzen das Angebotsintervall (Tabelle KAZY im SAP System) und die Kapazitätsbedarfe (Tabelle KAPA im SAP System) ein. Der ausgewählte Bereich wird gelöscht und es wird überprüft, ob weitere Angebotsintervalle und Kapazitätsangebote angepasst werden müssen. So muss beispielsweise immer mindestens ein Kapazitätsangebot im Form eines Standardangebotes vorhanden sein. Falls dieses fehlt wird es mit dieser Methode eingefügt. Falls Lücken zwischen den Gültigkeitszeiträumen der Angebotsintervalle entstehen, müssen ebenfalls Standardangebote eingefügt werden. Außerdem müssen eventuell die „Ränderwerte“ in ihrem Gültigkeitsdatum angepasst werden.
- `ANGEBOTSINTERVAL_CREATE_ENTITY` POST Methode für das Angebotsintervall. Diese werden mit dem Funktionsbaustein `/ISDFPS/WCUPS_WC_SAVE_UPDATE_T` verbucht.
- `ARBEITSPLATZSET_GET_ENTITY` Abrufen einer einzelnen Arbeitsplatzentität. Zu dem Arbeitsplatz werden zusätzlich noch Bezeichnung von Werk, Arbeitsplatz und Arbeitsplatzart aus weiteren Tabellen selektiert.
- `ARBEITSPLATZSET_GET_ENTITYSET` Abrufen aller Arbeitsplätze mit zusätzlichen Filterfunktionen für die Abfrageparameter Werk und Arbeitsplatzart. Mit dieser Methode wird auch die sortierte Ausgabe der Arbeitsplätze realisiert.
- `KAPAZITAETSANGEB_CREATE_ENTITY` Methode um neuen Kapazitätsangebote zu erstellen. Kapazitätsangebote werden mit dem gleichen Funktionsbaustein wie zur Erstellung der Angebotsintervalle verbucht.
- `KAPAZITAETSANGEB_GET_ENTITY` Abrufen eines einzelnen Kapazitätsangebots.
- `KAPAZITAETSANGEB_GET_ENTITYSET` Abrufen aller Kapazitätsangebote mit zusätzlichen Filterfunktionen für die Abfrageparameter zur Eingrenzung der übergeordneten Kapazität
- `KAPAZITAETSET_GET_ENTITY` Abrufen einer einzelnen Kapazität.
- `KAPAZITAETSET_GET_ENTITYSET` Abrufen aller Kapazitäten mit Filterfunktionen für den Aufruf über die „expand“ Funktion des Arbeitsplatzes.

- SCHICHTSET_GET_ENTITY Abrufen einer einzelnen Schichtdefinition.
- SCHICHTSET_GET_ENTITYSET Abrufen aller Schichtdefinition mit Filteroptionen für die Auswahl der zugehörigen Kapazität.

VII. APPLIKATION

A. Konzept

SAPUI5 wird in der Version 1.30 verwendet. Die SAPUI5 UI Controls werden so ausgewählt, dass ihre Bibliotheken miteinander kompatibel sind und ihre Elemente auch auf mobilen Geräten genutzt werden können. Durch die Auswahl der Controls anhand von UI-Designpatterns und Floorplans ist ein Rahmen gegeben, der Orientierung für den strukturellen Aufbau der Anwendung bietet. So ist bereits festgelegt, dass die Haupt View in zwei Teile, der „MasterPage“ und der „DetailPage“ aufgeteilt ist. Aufgrund der Komplexität der anzuzeigenden Tabelle (sie wird anhand der vorhandenen Schichtdefinitionen dynamisch erstellt), wird diese ebenfalls als eigene Komponente realisiert. Für die Datenhaltung und die Logik für die Abfragezeitraumerstellung und für die Komprimierung der Kapazitätsangebote und Angebotsintervalle wird ein eigenes Model namens „ShiftData“ erstellt. Außerdem existiert eine View, die eine Rückwärtsnavigation ermöglicht, falls eine nicht vorhandene Ressource der App aufgerufen wird. Zusätzlich mit dem Übersetzungsmodell („i18n“) und der Anbindung zum erstellten OData Service („default“) ergibt sich der in Abbildung 10 gezeigte Grundaufbau. Die Applikation wird über „Component.js“ als Komponente in ein SAP Fiori Launchpad eingebettet. Alle Controller erben zusätzlich von einer Basis Klasse („BaseController“). Über diesen erhalten sie die Standard Navigation Funktionen des Browser. Das Shiftdata Model erbt von einer Basisklasse („QueryPeriod“) die Fähigkeit, aus zwei Datumswerten ein Intervall der Tage zwischen diesen Werten, zu bilden.

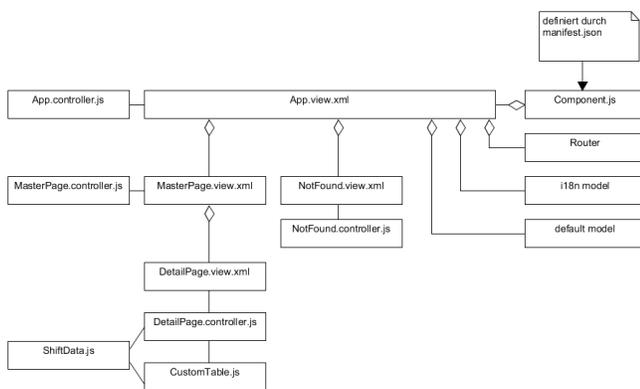


Abbildung 11. Grundaufbau „Schichteditor“

B. Implementierung

Bei der Implementierung entstanden viele Quelldateien. Nachfolgend wird bei den wichtigsten kurz auf ihre Funktion und etwaige Besonderheiten in ihrer Programmierung eingegangen.

- Bootstrapping: index.html Als Einstieg in die Webapplikation dient eine HTML Datei. Über diese wird das Kern Framework mit Parametern gestartet. Während der Initialisierung des Frameworks wird die Anwendung über die Component.js in das Fiori Launchpad gehängt. Alle durch Javascript erstellten visuellen Elemente werden in den Dokumentenkörper dieser HTML Datei eingefügt.
- Component, Deskriptor: Component.js, manifest.json Mithilfe von Components werden UI5 Webapplikationen unabhängiger von dem Fiori Launchpad. Inspiriert von dem Web-App-Manifest-Konzept (W3C) ermöglicht diese Klasse, dass die Webapplikation durch Deskriptoren definierte, zusätzliche Metainformationen besitzen können. Apps können dadurch auf „Startbildschirmen“ platziert werden. Zusätzlich wird mithilfe dieser Klasse das Routing, die gültige Datenquellen und Modelle für die gesamte App definiert. Der für die Navigation zuständige Router wird initialisiert und durch die Routingtabelle des Manifest wird festgelegt, welche von den vorhandenen Views die Hauptview ist. Deren Controller wird als erstes geladen. Ein falscher URL-Pfad im App-Namensraums führt zum „not-Found“ View.
- App View und Controller Der App Controller erstellt das Wurzelement der mobilen UI5 Anwendung. Die URL wird modifiziert (über Router Klasse) und anschließend wird zu der MasterPage View navigiert.
- MasterPage View und Controller Die MasterPage View erstellt das Suchfeld für die Liste aller Arbeitsplätze. Außerdem generiert sie eine Toolbar mit Buttons für die Sortier- und Gruppierfunktion dieser Liste. Der zugehörige Controller lädt aus JSON Dateien die Merkmale, nach denen die angezeigte Liste sortiert und gruppiert werden kann. Für die Gruppierung der Liste besitzt der Controller eine Factory Method, welche ein Controller-Template für die Erstellung der Gruppenköpfe aus-prägt. Außerdem werden mit diesem Controller die Suche (Filteroperation) in dem als Model angebindenen OData Service nach Arbeitsplatz und Werk durchgeführt. Wenn eine Auswahl eines Arbeitsplatzes stattfindet, informiert der Controller über ein EventBus-Entwurfsmuster den DetailPage Controller über die Arbeitsplatzauswahl.
- DetailPage View und Controller Der Detailpage View erstellt die Object Page und eine Toolbar mit Buttons für die Bearbeitung der Tabelle. Im Controller wird am EventBus gelauscht ob ein Arbeitsplatz ausgewählt wurde. Sobald dies passiert ist, werden die Kapazitäten zu dem Arbeitsplatz abgefragt und in Tabs dargestellt. Für die direkte Navigation auf den ersten Tab, wird in den Lebenszyklus des ObjectPage Controls eingegriffen. Es wird überprüft ob es zu der, durch den Tab repräsentierte

Kapazität, eine aktive Version und Schichtdefinitionen gibt. Falls keine existieren oder keine Kapazität aktiv ist, werden Hinweistexte ein-geblendet. Falls eine aktiven Version vorhanden ist, wird eine Datumsauswahlleiste und das ShiftData Model erstellt. Die Datumsauswahl wird per One-Way Binding an dieses Model gebunden. Damit führen Änderungen bei der Datumsauswahl direkt zu einer Änderung des Models. Mithilfe der Basis-klasse werden Intervalle aus Tagen gebildet. Schlussendlich wird die Tabelle aufgebaut. Diese nutzt ebenfalls das ShiftData Model zur Datenhaltung. Die Tabelle ist ein eigenes Control vom Typ CustomTable. Der DetailPage Controller kümmert sich zusätzlich darum, dass die Bearbeitungsbefehle ("Bearbeiten"/ "Abbrechen"/ "Speichern") entsprechend des Tabellenstatus ein-geblendet werden. Die Befehle werden an das ShiftData Model und CustomTable Control weitergereicht.

- Tabelle: CustomTable.js Die Tabelle wird dynamisch aufgebaut. Die Zeilenanzahl wird durch die Anzahl der Tage in dem Abfragezeitraum bestimmt. Die Spaltenanzahl wird durch die Anzahl der verschiedenen, zu der Kapazität gepflegten, Schichtdefinitionen bestimmt. Mit der Tabelle wird das Kapazitätsangebot der Kapazität gepflegt. Wenn im System keine Schichtdefinitionen verwendet wurden, sondern das Kapazitätsangebot manuell gepflegt wurden, dann erkennt die Tabelle dies und schließt diese Zeile von der Bearbeitung aus. Zudem hat die Tabelle eine Eingabeüberprüfung. Es können in der Zeile keine Schichten ausgewählt werden, die sich zeitlich mit anderen überschneiden. Diese Funktionen werden mit sogenannten "Formatter" Funktionen realisiert. Das Control arbeitet mit dem ShiftDataModel.
- ShiftData.js Das ShiftData Model ist zuständig dafür, mithilfe des OData Service Schichtdefinitionen, Angebotsintervalle und Kapazitätsangebote einer Kapazität vom Backendsystem abzufragen, zu verwalten und beim Speichern in einer passenden Form wieder im Backendsystem zu speichern. Es übernimmt auch die nötigen Formatierungen, um diese Daten in einer Tabelle anzeigen zu können. Die Schichtdefinitionen einer Kapazität müssen für die Eingabeüberprüfung der auf Überschneidungen überprüft werden. Zusätzlich übernimmt das ShiftData Model die Komprimierung der Kapazitätsangebote für die Speicherung über den Service.

VIII. ERGEBNIS UND BEWERTUNG

Das Ergebnis dieser Arbeit ist eine lauffähige Applikation, welche nach der Fiori UX Strategie entwickelt wurde. Diese Applikation kann sowohl mit einem Desktop PC als auch mit einem Tablet bedient werden. Mit der Applikation lassen sich sehr einfach und übersichtlich Schichten zu einzelnen Arbeitsplätzen bearbeiten. Es kann innerhalb der Anwendung leicht zwischen den einzelnen zu bearbeitenden Kapazitäten navigiert werden und die Bedienung ist intuitiv. Der Nutzer erhält einen schnellen Überblick über die Kapazitätsauslastung

des Arbeitsplatzes und er kann sich sofort die Bedarfsverursacher anzeigen lassen. Der Anwender kann keine falschen Eingaben tätigen und er erhält sofort Rückmeldung über die Auswirkung seiner Auswahl. Neben der Einsparung von Aktionen bietet die Applikation eine deutliche Verringerung der Komplexität. Durch die Applikation kann innerhalb eines Bildschirmbildes verschiedene Arbeitsplätze bearbeitet werden. Die SAP Anwendung ist nicht intuitiv. Anhand eines Datums muss der Nutzer herausfinden, welchen Wochentag er in welchem Intervall bearbeiten muss. In der erstellten Applikation muss er diese Denkleistungen nicht voll-bringen. In Abbildung 12 ist dargestellt wie die Schichtänderung mit der neuen Applikation funktioniert. Innerhalb von drei Aktionen können einfach Schichten hinzugefügt oder gelöscht werden. Zusätzlich gibt es eine Balkenanzeige über die Auswirkungen der Auswahl. Für eine bessere Verteilung der App würde

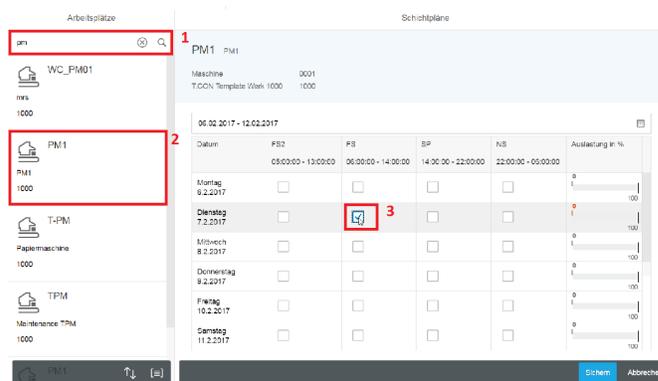


Abbildung 12. Schichtänderung mit der neuen Applikation

es sich auch anbieten den zugehörigen Service zu "Central Hub Deployment - Development on the Hub" zu modifizieren. Dadurch würde der Service mobiler und könnte in allen Fiori Landschaften eingesetzt werden.

LITERATUR

- [1] Mathew, Bince. 2015. *Beginning SAP Fiori*. New York: Apress.
- [2] Dickersbach, Jörg Thomas and Keller, Gerhard. 2010. *Produktionsplanung und -steuerung mit SAP ERP: Grundlagen - Prozesse - Customizingwissen*. Bonn: Rheinwerk Verlag and SAP Press.
- [3] Englbrecht, Michael and Wegelin, Michael. 2015. *SAP Fiori: Implementierung und Entwicklung*. Bonn: Rheinwerk Verlag and SAP Press.
- [4] Bönnen, Carsten and Drees, Volker and Fischer, André and Heinz, Ludwig and Strothmann, Karsten. 2014. *OData und SAP Gateway*. Bonn: Rheinwerk Verlag and SAP Press.

IX. ANHANG

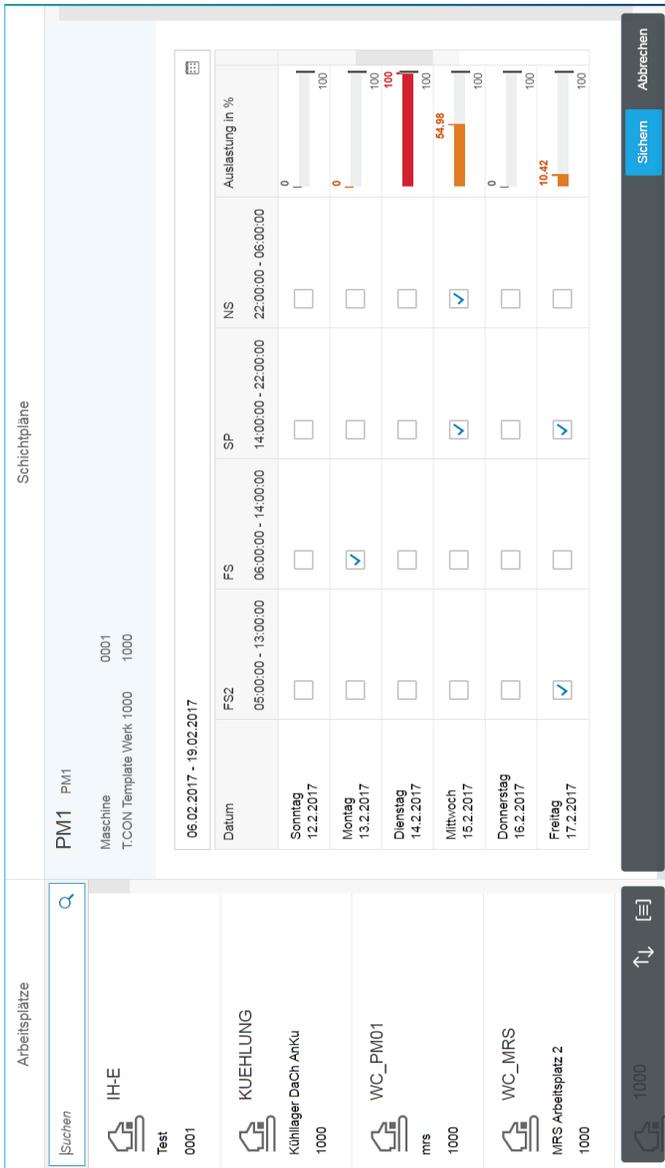


Abbildung 13. GUI der „Schichteditor“ Applikation

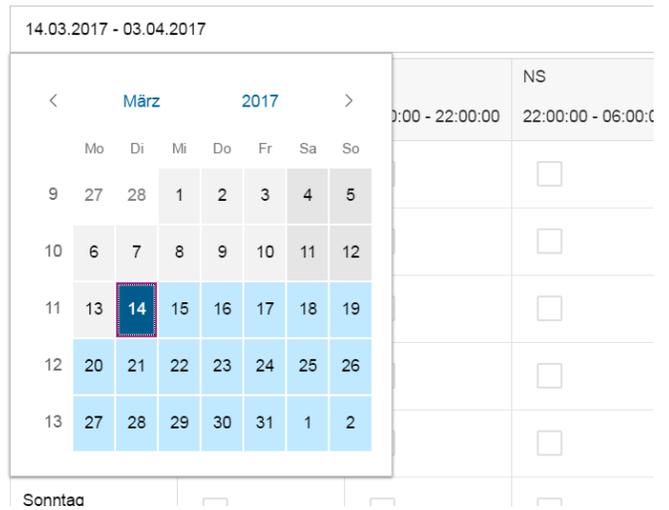


Abbildung 14. Abfragezeitraumauswahl

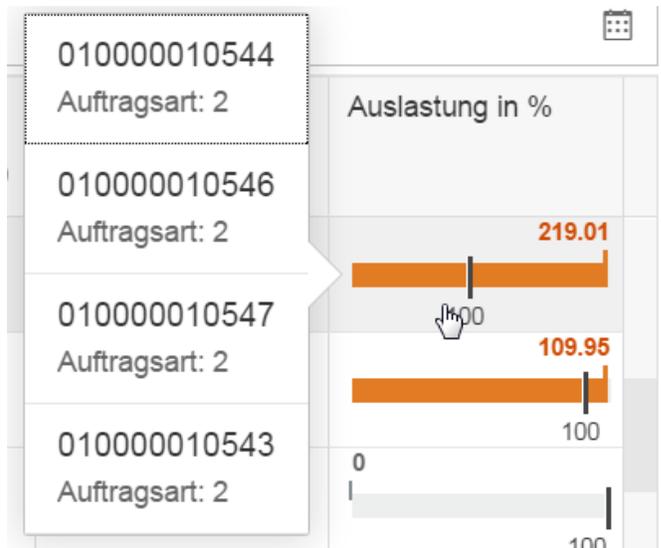


Abbildung 15. Kapazitätsbedarfsverursacher an einem ausgewählten Tag

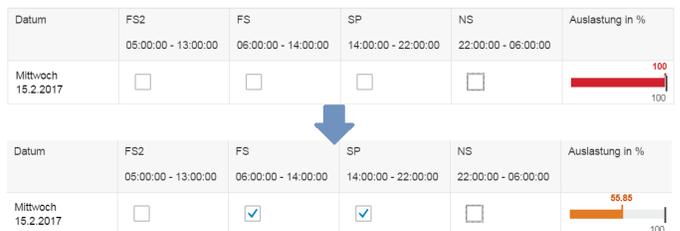


Abbildung 16. Veränderung der Arbeitsplatzauslastung bei Schichtänderung