

Von der Applikationsentwicklung zum Toolboxlieferanten – ein Erfahrungsbericht

Dr. Markus Kubik / Wolfgang Vonnemann
METRO SYSTEMS GmbH
Düsseldorf

Schlüsselworte

Oracle DB, BPM, Messaging, Java, Anwendungsentwicklung, Rollen, Erfahrungsbericht, ext JS, BIRT.

Einleitung

Das Produkt Metro Supply Logistic System, kurz MSLS, ist eine Java Rich-Client Anwendung zur Abrechnung extern bezogener Logistikdienstleistungen. Diese wurde seitens METRO SYSTEMS für die konzerninterne Logistikfirma METRO LOGISTICS entwickelt und ist seit 8 Jahren in mittlerweile 31 Vertriebslinien-Länder-Kombinationen innerhalb der METRO-Welt im Einsatz. Mit jeder neuen Version wurden Demands zukünftiger Endanwender oder bereits aktiver User umgesetzt – was zu einem sehr komplexen und starrem System führte. Die daraus resultierenden Probleme hinsichtlich der Überschaubarkeit der Anwendung für den Anwender und der schwieriger werdenden Wartbarkeit durch die Entwicklung gaben den Anlass über ein Redesign nachzudenken und dabei gemachte Erfahrungen aus den Vorjahren als auch Forderungen seitens der Fachbereiche hinsichtlich Kosteneffizienz und Umsetzungsgeschwindigkeit mit einzubeziehen. Passend zu den Anforderungen sollte das neue Produkt MSLS Express lauten.

Kritikpunkte an MSLS

MSLS ist eine typische Java Rich-Client Anwendung (RCP), deren GUI sich kaum von anderen RCPs unterscheidet und zumindest vom Design her „in die Jahre gekommen“ ist:

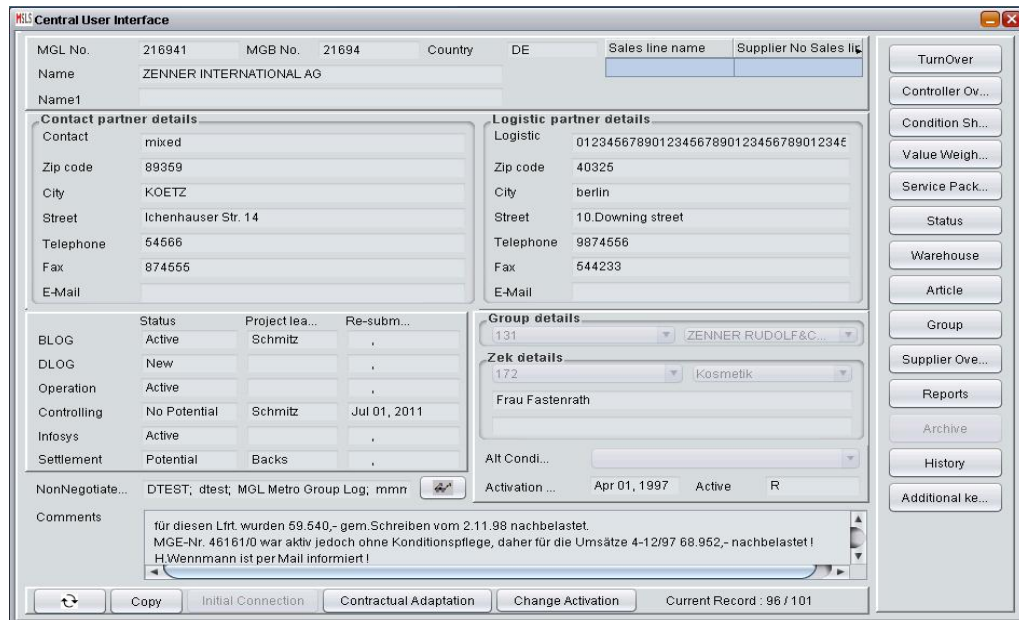


Abb. 1: Beispielmaske aus MSLS Stammdaten.

Die über 230 Screens umfassende Applikation musste sich folgenden Kritikpunkten stellen:

- Starrer Aufbau
Aufbau der Masken sind vom Fachbereich vorgegeben und können von einem Anwender nicht geändert werden, auch wenn die Platzierung einzelner Elemente für ein Land / eine Vertriebslinie / einen Anwender nicht sinnvoll ist. Eine spätere Änderung oder Hinzunahme von Funktionen hat einen nicht vertretbaren Aufwand zur Folge
- Dialoge überladen
Um möglichst allen Anwendern gerecht zu werden wurden alle jemals geforderten Funktionalitäten ohne Konfigurierbarkeit implementiert. Anwender, deren Organisationen zur Abbildung ihrer Geschäftsprozesse nur einen Bruchteil der angebotenen Funktionen benötigen, sind von der Anwendung erschlagen und benötigen hohen Schulungsaufwand, der zusätzliche Kosten verursacht.
- Keine Onlinehilfe
Auf eine Onlinehilfe wurde beim Erstellen der Applikation zugunsten Benutzerhandbücher verzichtet. In der Regel sind die Schulungsaufwände sehr hoch.
- Textbasierte Darstellung von Daten
Eine Analyse von wertvollen Daten aus dem Transportlogistikbereich z. B. ist nur über den komplizierten Weg von Exporten und Importen in andere Werkzeuge möglich.
- Geographische Transportmodelle werden nur über Tabellen abgebildet
Heutige state-of-the-art Möglichkeiten (maps) werden nicht angeboten.
- Arbeiten nur am PC möglich
Die Möglichkeit, Arbeiten mobil ohne festinstallierte Software zu erledigen, ist eine der steigenden Anforderungen.

Zusammengefasst wird die Applikation als zu alt, zu starr und letztlich als zu teuer empfunden.

Anforderungen an MSLS Express

Ein neues System sollte folgende Anforderungen berücksichtigen und damit die Notwendigkeit einer Investition eines Redesigns von MSLS rechtfertigen:

- Dialoge flexible / konfigurierbar für User
Pro Organisationseinheit sollen nur die Funktionen dem Anwender zur Verfügung stehen, die von ihm zur Erledigung des Geschäftes benötigt wird. Dazu soll die Anordnung von Pflege- und Anzeige-Items in enger Abstimmung zwischen Fachbereich und Anwender erfolgen. Die Folge ist eine hohe Akzeptanz der Applikation bei den Anwendern als auch ein geringerer Schulungsaufwand. Durch die Konfigurierbarkeit soll eine schnellere und günstigere Umsetzung von Anwenderanforderungen gewährleistet werden.
- Moderner Stil der Oberfläche
Die Anwender von heute unterscheiden sich stark von den Anwendern von vor 10 Jahren. Internetapplikationen wie amazon.de oder ebay.de sowie Apps für Android oder iOS zeigen den Anwendern von heute was technisch möglich ist.

Für eine möglichst hohe Akzeptanz bei den Anwendern wird daher eine moderne Oberfläche gefordert.

- Integrierte Onlinehilfe (WiKi)
Die Onlinehilfe soll den Anwender bei seiner täglichen Arbeit direkt unterstützen (kontextsensitiv). Die Bedienbarkeit der Onlinehilfe soll sich ihm selbst erschließen bzw. von bekannten Webseiten wie wikipedia.de ableiten lassen. Ziel ist eine Verringerung von Schulungsaufwand und –kosten durch Hilfe zur Selbsthilfe.
- Grafische Aufbereitung von Daten
Zur Unterstützung von Analysetätigkeiten sollen grafische Auswertungen ermöglicht werden. Diese haben den Vorteil, dass sie schneller erfasst werden können. Zudem ermöglichen Grafiken, die parallel zu den eigentlichen Basisdaten dargestellt werden, eine gleichzeitige Betrachtung unterschiedlicher Faktoren.
- Kartographische Unterstützung
Bei der Analyse von Warentransporten ist eine kartographische Unterstützung wichtig, da hierdurch eine geografische Visualisierung der Startpunkte, Zielpunkte und Wege in Bezug zu weiteren Größen wie Warenwert, Warengewicht, usw. erfolgen kann. Dem Nutzer wird dadurch ein zusätzliches Werkzeug gegeben um bspw. Transportwege noch weiter zu optimieren.
- Mobiles Arbeiten ermöglichen
Anwender von heute sind es durch verfügbare Smartphones oder Tablets gewohnt, bislang nur stationär durchgeführte Tätigkeiten auch mobil zu erledigen. Auch die neue Applikation muss dem Trend des mobilen Arbeitens folgen.

Zusammengefasst soll die neue Applikation flexibel und modern sein sowie eine günstige Weiterentwicklung und Konfiguration ermöglichen. Die Aufgabe der Entwicklung wird in der Bereitstellung von Komponenten zur Erstellung von Applikationen gesehen.

Umsetzung Teil 1 – Änderung der Systemarchitektur

Die Architektur der Applikation MSLS ist in Abbildung 2 wiedergegeben. Es handelt sich um eine Drei-Schichten-Architektur mit einer Datenhaltungsschicht in Form der Oracle-Datenbank, einer Logikschicht in Form des Oracle iAS und einer Präsentationsschicht unter Benutzung von Swing UI Komponenten.

Die Businesslogik ist zum Teil innerhalb der Datenbank und zum Teil innerhalb der Applikation implementiert. Die Verbindung der Applikation zur Datenbank erfolgt mit dem Object-Relation-Mapper TopLink.

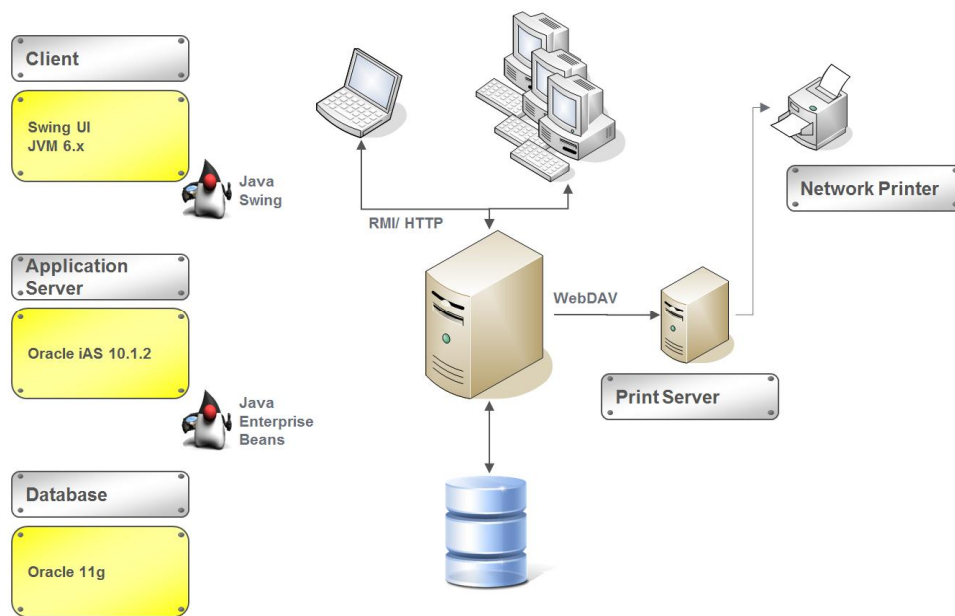


Abb. 2: Systemarchitektur MSLS.

Die neue Systemarchitektur (Abb. 3) umfasst ebenfalls eine Drei-Schicht-Architektur. Als Datenhaltungsschicht werden Oracle-Datenbanken eingesetzt. Auch mySql- oder H2-Datenbanken können genutzt werden.

Die Art und Anzahl der benutzten Datenbanken ist per Konfiguration einstellbar. Ein OR-Mapper kommt nicht zum Einsatz. Die Verbindung der Kern-Applikation zur Datenbank erfolgt per JDBC.

Der Applikations-Server (Apache Tomcat) stellt die Kern-Applikation zur Verfügung. Die Kern-Applikation enthält keine Businesslogik. Die Kern-Applikation erfüllt zwei wesentliche Aufgaben

- Auslieferung von Dialog-Komponenten-Beschreibungen
- Vermittlung zu den umgebenden Systemen

Um die Unabhängigkeit der umgebenden Systeme zu gewährleisten, kommuniziert die Kern-Applikation ausschließlich mit einem Messaging-System (Apache ActiveMQ).

Die umgebenden Systeme liefern folgenden Funktionen:

- | | |
|---------------|--|
| Reports | „Business Intelligent and Reporting Tool (BIRT)“ |
| Excel-Exports | „Business Intelligent and Reporting Tool (BIRT)“ |
| ETL | „Pantaho Kettle“ |
| BPM | „Activiti BPM Platform“ |

Die Businesslogik des Gesamtsystems wird an die BPM delegiert.

Als Client kommt ein Webbrowser zum Einsatz (HTML 5).

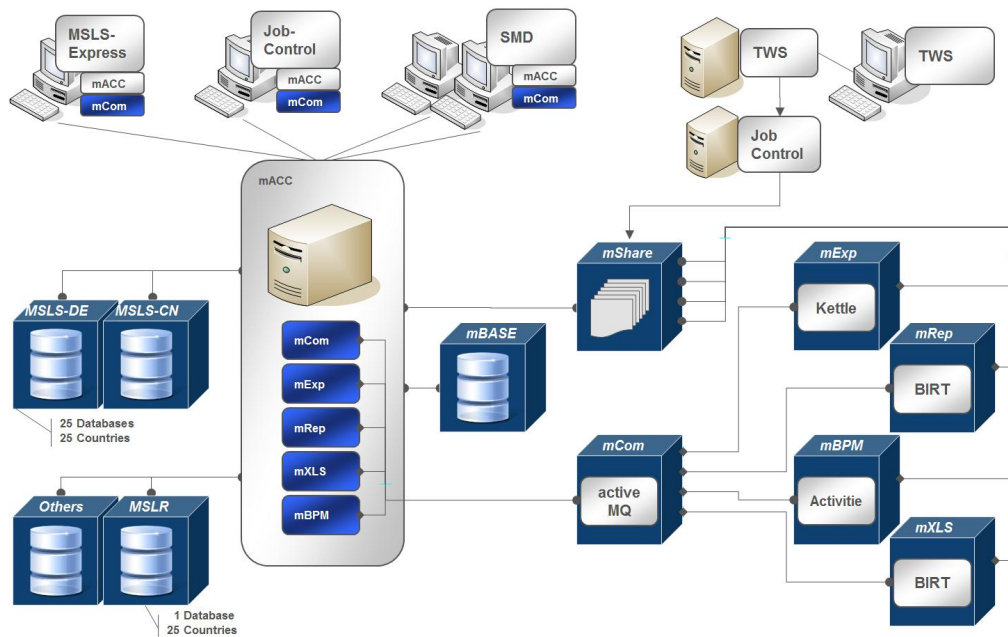


Abb. 3: Systemarchitektur MSLS Express.

Die Entwicklungstätigkeit besteht nicht mehr darin, angeforderte Applikationen zu erstellen bzw. Änderungen an Applikationen durchzuführen, sondern Komponenten (tools) zu entwickeln, die per Konfiguration durch den Fachbereich in enger Abstimmung mit den Endanwendern auf eine Oberfläche gebracht werden können.

Umsetzung Teil 2 – Änderungen beim Entwicklungsprozess und bei den Rollendefinitionen

Neben den geschilderten Kritikpunkten an der Altapplikation MSLS (s. oben) hat die Analyse der bisherigen Vorgehensweise bei der Entwicklung neuer Funktionen ergeben, dass die Zeit zwischen Anforderung von der Endanwenderseite und Produktivsetzung als zu lang empfunden wurde.

Zudem besaß der bisherige Entwicklungsprozess (siehe Abb. 4), der die klare Trennung von Endanwender und Entwickler durch „Zwischenschalten“ des Fachbereiches vorsah, das Risiko, dass umgesetzte Anforderungen aufgrund geänderter Bedingungen auf Anwenderseite nicht mehr passten. Der Endanwender sah die umgesetzten Anforderungen frühestens im UAT (User Acceptance Test), nachdem die Entwicklungsschritte bereits durch funktionale und technische Spezifikationen abgeklärt und dokumentiert sowie durch die Qualitätssicherung (QA) abgenommen wurden.

Dies hatte zur Folge, dass zwar richtig, aber nicht immer das Richtige entwickelt wurde und neben personellen Ressourcen auch finanzielle Mittel zum Teil falsch eingesetzt wurden.

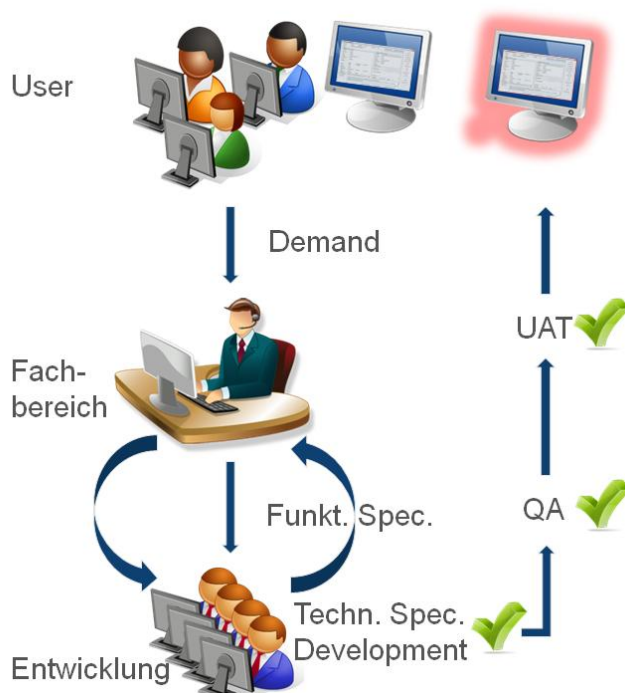


Abb. 4: Entwicklungsprozess bisher: ein Demand wird seitens der User an den Fachbereich gestellt. Dieser überlegt sich eine Lösung und formuliert diese in Form einer funktionalen Spezifikation an die Entwicklung. Die Entwicklung entwickelt daraus eine technische Spezifikation und stimmt sich mit dem Fachbereich ab. Nach Fertigstellung der Entwicklung wird die Neuentwicklung funktional und integrativ durch eine Qualitätssicherungsstufe (QA) getestet, bevor sie den Anwendern im Rahmen eines user acceptance test (UAT) zur Verfügung gestellt wird. Bei erfolgreicher Abnahme erfolgt die Produktivnahme.

Das neue Konzept (Abb. 5) sieht vor, dass nach Stellen eines Demands seitens der Anwender der Fachbereich eine Lösung durch Konfiguration vorhandener Komponenten erstellt. Die Aufgabe der Entwickler besteht nicht mehr darin, komplett spezifizierte Applikationen zu entwickeln, sondern Komponenten zu programmieren, die später in neue oder bestehende Oberflächen eingebunden werden können. Das Zusammenspiel dieser tools nach erfolgreicher Konfiguration stellt die Applikation dar.

Konfiguriert werden die Applikationen durch den Fachbereich, der dieses in sehr enger Abstimmung mit den Anwendern durchführt. Eine wöchentliche Videokonferenz mit Live-Demo und Abgleichung von Erwartungen hat sich bei dem Teilmodul SMD (Supplier Master Data) bewährt. Die zweite und langfristig überwiegende Aufgabe der Entwickler ist es, dem Fachbereich als Berater bei der Durchführung der Konfiguration der Komponenten zur Seite zu stehen.

Dieses Konzept erfordert von allen Beteiligten eine geänderte Rollenwahrnehmung: Der Anwender muss sich jetzt aktiv in den Prozess der Applikationskonfiguration einbringen. Der Fachbereich muss Wissen über die Möglichkeit und Konfigurierbarkeit zur Verfügung stehender Komponenten aufbauen und einsetzen sowie die Gestaltung der zukünftigen Applikation aktiv gestalten. Unterstützt wird er durch die Entwicklung auf zwei Arten: auf der einen Seite durch das Zurverfügungstellen neuer Komponenten – hier programmiert die Entwicklung; auf der anderen Seite durch ausgiebige Beratung bei der Konfiguration von Komponenten zur Umsetzung der Anwenderanforderungen.

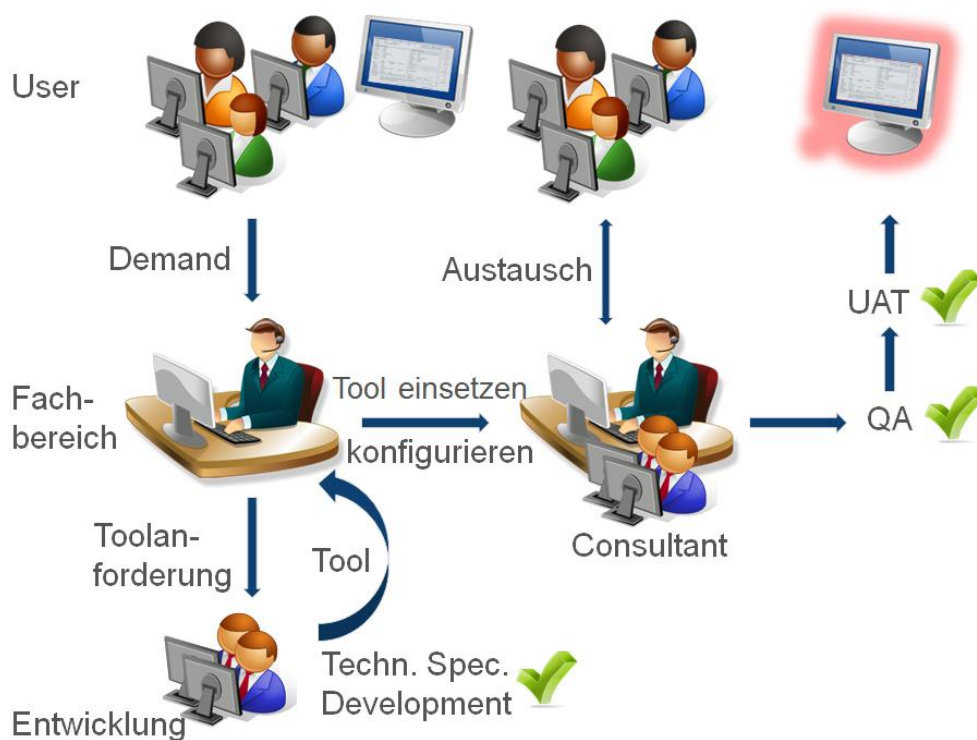


Abb. 5: Neuer Entwicklungsprozess: ein Demand wird seitens der User an den Fachbereich gestellt. Dieser überlegt sich wie eine Lösung mit bestehenden Komponenten umzusetzen bzw. abzubilden ist. In regem Austausch mit den Anwendern wird die finale Lösung abgestimmt und konfiguriert. Nach einer QA-Phase folgt die nun kürzere UAT-Phase, da die Endanwender bereits Großteile der neuen Funktion kennen. Nach erfolgter Freigabe folgt die Produktivnahme. Die früheren Entwickler stehen nun dem Fachbereich als Berater zur Seite und unterstützen bei der Konfiguration. Sollte für die Erstellung einer applikatorischen Lösung eine neue Komponente benötigt werden, so wird diese Anforderung an die Entwicklung kommuniziert und von ihr entwickelt und zur Verfügung gestellt. Der Anteil der reinen Entwicklertätigkeit verlagert sich langfristig in Richtung Beratung – eine komplette Applikationserstellung findet durch die Entwicklung nicht mehr statt.

Erste Erfahrungen / Rückmeldungen

Als erster Schritt Richtung Redesign von MSLS wurde das Thema Stammdaten angegangen. Neben den Grundfunktionalitäten („Kernel“) wurden nach Abstimmung seitens des Fachbereichs mit den Endanwendern Komponenten wie „Fileupload“ oder „Filterfunktion“ durch die Entwicklung programmiert. Nach der Schulung des Fachbereiches wurde eine erste Version der geforderten neuen Module durch diesen erstellt und in wöchentlichen Videositzungen mit den Endanwendern diskutiert, optimiert und schließlich als Teilmodul SMD (Supplier Master Data, Abb. 6) produktiv gesetzt.

Die Vorgehensweise kam vor allem beim Endanwender sehr gut an; er fühlte sich abgeholt, verstanden und war über die schnelle Reaktionszeit bei der Umsetzung seiner Vorstellungen erfreut. Die Einbeziehung der Entwickler in die wöchentlichen Videositzungen hatte den Vorteil, dass einerseits Verständnis für die geschäftlichen Belange seitens der Entwickler geschaffen wurde andererseits auch eine rasche Klärung/Erklärung auf Seiten der Anwender möglich wurde, wenn sich Anforderungen technisch nicht umsetzen ließen. Der Fachbereich lobte die enge Zusammenarbeit und die

Möglichkeit bei der Gestaltung der Lösung für den Anwender aktiv mitzuarbeiten. Alles in allem wurde von allen Seiten positiv vermerkt, dass der Endanwender die für ihn richtige Lösung schneller und günstiger als früher bekommt und dies zusätzlich auf einer modernen technischen Plattform.

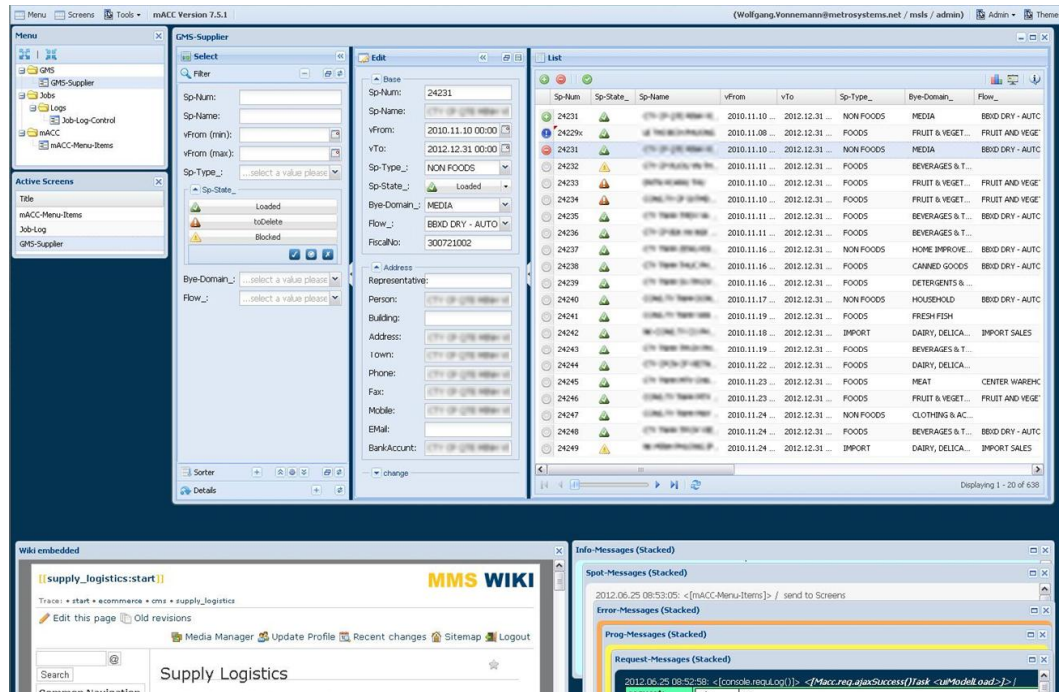


Abb. 6: Neue Screens in MSLs Express – hier ein Ausschnitt aus SMD (Supplier Master Data) im oberen Abschnitt. Im unteren Abschnitt ist die Onlinehilfe in Form eines Wiki sowie diverse Messagetypen erkennbar.

Fazit

Die Forderungen seitens der Endanwender an das Produkt MSLs nach höherer Flexibilität und Modernisierung wurden durch Änderung der Systemarchitektur, aber auch durch eine neue Entwicklungsstrategie und verändertes Rollenverständnis erreicht. Auch wenn sich das neue Konzept noch über die Jahre bewähren muss, ist der erste Schritt beim Redesign der betrachteten Applikation erfolgreich getan.

Kontaktadresse:

Dr. Markus Kubik
METRO SYSTEMS GmbH
Metro-Straße 12
D-40235 Düsseldorf

Wolfgang Vonnemann
METRO SYSTEMS GmbH
Metro-Straße 12
D-40235 Düsseldorf

Telefon: +49 (0) 211-969-4228
Fax: +49 (0) 211-969-2939
E-Mail markus.kubik@metrosystems.net
Internet: www.metrosystems.net

+49 (0) 211-969-1644
+49 (0) 211-969-2939
wolfgang.vonnemann@metrosystems.net
www.metrosystems.net