Sicherer Hafen

Ein Leitstand mit Java für Hamburgs „Herzkammer“

Holger Breitling, Arne Scharping

Damit ein Schiff vom Meer aus sicher und zügig die Elbe hinauf und in den Hamburger Hafen gelangen kann, müssen die Mitarbeiter der Nautischen Zentrale der Hamburg Port Authority (HPA) zahlreiche Faktoren im Blick behalten: etwa andere Schiffe, Baustellen, Pegelstände, Durchfahrtshöhen oder Liegeplätze. Seit Herbst 2012 unterstützt sie dabei ein in Java geschriebenes und auf Eclipse RCP, Spring und Webservices basierendes Leitstandssystem: der „Integrierte Port Monitor“, der die Informationen aus den vielen bereits vorhandenen Einzelsystemen zusammenführt und so besser handhabbar macht. In diesem Beitrag stellen wir die Architektur des Port Monitors und seine Entwicklungsgeschichte vor.

Der *Integrierte Port Monitor* baut auf drei grundlegenden Ideen auf:

- der kartenbasierten, georeferenzierten Integration von Informationen auf der Oberfläche,
- einem Arbeitsplatzkonzept mit verschiedenen Port-Monitor-Clients, die für die Arbeitsplätze und Aufgaben angepasst sind, sowie
- der minimal-invasiven Integration der bestehenden Systemlandschaft.

Vom Forschungsprojekt zur Leitstandsplattform

Der Integrierte Port Monitor nahm seinen Ausgang im Forschungsprojekt *GeneAL* [GeneAL]: Das von C1 WPS und dem Fachbereich Informatik der Universität Hamburg durchgeführte und von der Innovationsstiftung Hamburg (heute: Hamburger Investitions- und Förderbank) geförderte Forschungsprojekt soll grundlegende Architekturbestandteile und Komponenten von Leitständen identifizieren und generisch wiederverwendbar implementieren. Bei der HPA, einem von drei GeneAL-Industriepartnern, konnte das Projekt von Vorarbeiten durch *AHOI* [AHOI] profitieren, einem anderen Forschungsprojekt in der Nautischen Zentrale.

Im GeneAL-Projekt entstand ein Prototyp für das spätere Kartenwerkzeug des Port Monitors. Die Nautiker bewerteten das Ergebnis so positiv, dass ein kommerzielles Anschlussprojekt zustande kam, in dem der Prototyp schließlich zur produktiven Anwendung ausgebaut wurde.

Alles Geo

Vorbild für das Kartenwerkzeug war der Umgang der Nautiker mit ihrer großen Wandkarte. Diese Karte des Hamburger Hafens war wandfüllend an einer Seite der Nautischen Zentrale angebracht. Die Nautiker benutzten sie zur Vergegenwärtigung aktueller oder geplanter Maßnahmen und Zustände im Hafengebiet: Sie hefteten Magnetschiffe auf der Karte an, vermerkten Baustellen und weitere Geschehnisse mit Hilfe von passenden Stecksymbolen. Zu diesem Zweck standen sie regelmäßig von ihrem Arbeitsplatz – an dem sie mit mehreren Monitoren jeweils getrennte Informationssysteme überwachten – auf, um den Stand an der großen Wandkarte zu aktualisieren. Der Bedarf nach einem kartenbezogenen Informationssystem war bei dem Anblick „mit Händen zu greifen“.

Das Kartenwerkzeug ist der Kern dieses kartenbezogenen Informationssystems. Für die Darstellung der Karte wird das bei Behörden übliche UTM-Koordinatensystem genutzt. Viele der verfügbaren Darstellungswerkzeuge auf der Grundlage von Internet-Diensten wie Google Maps und OpenStreetMap basieren anders als „Universal Transverse Mercator“ [UTM] auf der einfacheren Mercator-Projektion und kommen daher nicht in Frage. Auf der elektronischen Karte werden die bisher manuell auf der Wandkarte gepflegten Informationen (und viele weitere) sichtbar gemacht, und zwar georeferenziert – also ortsbezogen. Die wichtigsten im Kartenwerkzeug dargestellten Informationen umfassen (in Klammern die Herkunft):

- Schiffspositionen, die von den Schiffen selbst übertragen werden (AIS-Daten des Automatischen Identifikationssystems),
- Baustellen, Hindernisse, schifffahrtspolizeiliche Maßnahmen (Port Monitor bzw. Geoinformationssystem),
- Brückeninformationen mit Durchfahrtshöhen (Brücken-Datenbank),
- Pegelstände (Pegelstationen der HPA),
- Peildaten – zukünftig (Peilerei-System),
- Liegeplätze und deren Belegung (Informationssystem DV Elbe, Geoinformationssystem).

Die Nautiker können die eingeblendeten Symbole und Informationen in der Kartenansicht direkt manipulieren und den Systemzustand auf diese Weise unmittelbar aktuellen Ereignissen anpassen.

Architekturüberblick

Technologiestack

Der Port Monitor ist ein Rich Client auf Basis der Eclipse Rich Client Platform. Zusätzlich werden im Client auch Spring und Spring Dynamic Modules eingesetzt. Wesentliche Teile der Logik und Informationsbeschaffung realisiert der Client durch den Aufruf entfernter Services. Diese werden auf einem Server zur Verfügung gestellt und greifen von dort aus auf einzelne Backend-Systeme zu oder implementieren Logik „standalone“.

Die Services laufen im Spring-Container innerhalb eines Tomcat-Servlet-Containers und stehen überwiegend als Webservices zur Verfügung. Für die Bereitstellung als Webservice wird das CXF-Framework benutzt.

Statische Architektur

Die Aufteilung der Klassen in Plug-ins bzw. Module ergibt sich anhand zweier Dimensionen. In der *1. Dimension* wird nach Verteilung angeordnet. Damit ergeben sich die Ausprägungen (horizontale Schichten): Client, Server und Common. Common-Projekte:

- werden in den Client- und Server-Modulen genutzt (Materialien, Fachwerte, Service-Interfaces, einzelne Service-Implementationen).
- enthalten teilweise neben der puren Spring-Kontextdatei eine weitere Kontextdatei, die Spring-DM-Funktionalität (Im- und Export dynamischer Services) ergänzt und nur auf dem Client geladen wird.

In der *2. Dimension* wird anhand fachlicher Schnitte unterteilt, das heißt, es gibt vertikale Schichten zu den einzelnen fachli-

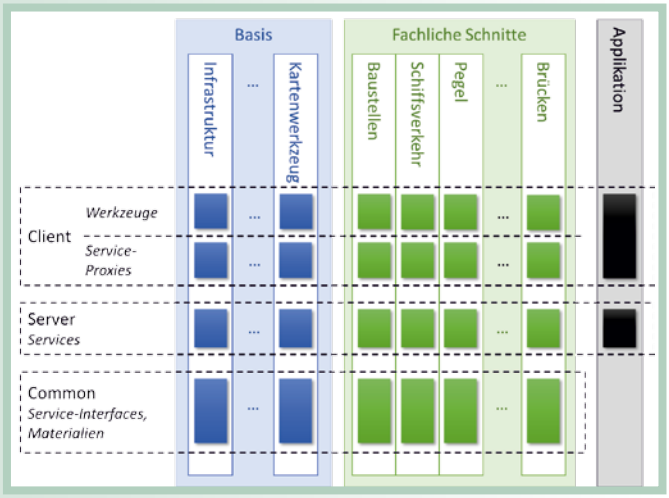

Abb. 1: Schnitt der Plug-ins/Module

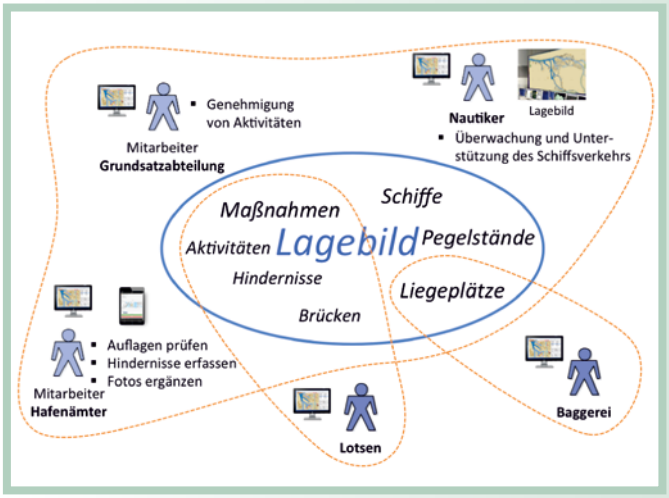

Abb. 2: Zentrale Konzepte, Arbeitsplätze und Aufgaben

chen Domänen. Zusätzlich gibt es technisch geprägte Projekte mit Basisfunktionalität und solche mit „generischen Bausteinen“.

Arbeitsplatzkonzeption und Eclipse RCP

Weil der Port Monitor eine sehr responsive und performante Benutzungsoberfläche verlangt, haben wir ihn als Rich Client implementiert. Eclipse RCP wurde als Implementierungstechnologie ausgewählt, weil die darin enthaltenen Konzepte Plug-ins, Extension-Points und Perspektiven sehr gut zu einer generischen Leitstandsplattform passen. Perspektiven stellen geeignete, zueinander passende Softwarewerkzeuge (Oberflächenelemente) für eine bestimmte Aufgabe zusammen und sind darum eine geeignete Vergegenständlichung für die verschiedenen Arbeitskontexte oder Arbeitsplätze, in denen der Port Monitor eingesetzt wird.

Der Port Monitor richtet sich neben den Nautikern auch an Arbeitsplätze in den Hafenämtern, in der Abteilung für Grundsatzangelegenheiten und absehbar an Schutendisponenten (es geht um die Schiffe, die beim Ausbaggern des Hafens und der Elbe eingesetzt werden). Weitere Arbeitsplätze sind geplant.

Perspektive	Aufgabe	Arbeitsplätze
Lagebild	Überblick über die Lage im Hafen	Alle
Verwaltung	Verwaltung von Hindernissen, Aktivitäten, schifffahrtspolizeilichen Maßnahmen	Naut. Zentrale, Hafenämter
Genehmigung	Genehmigung von Aktivitäten	Abt. für Grundsatzangelegenheiten
Administration Videowall	Administration der Videowall	Naut. Zentrale
Schutendisposition (geplant)	Planung und Steuerung von Baggereiaktivitäten	Schutendisponent

Tabelle 1: Perspektiven der Anwendergruppen

Die aktuell relevanten Perspektiven sind in Tabelle 1 zusammengefasst.

Eine Perspektive gibt einzelnen Anwendergruppen zielgenau nur abgestimmte, passende Werkzeuge an die Hand und ist Teil des Berechtigungskonzepts. Entsprechend verfügt etwa ein Mitarbeiter im Hafenamt nur über die Perspektiven „Lagebild“ und „Verwaltung“ und kann die anderen Perspektiven nicht nutzen.

Das Kartenwerkzeug

Der wichtigste Bestandteil des Port-Monitor-Clients ist das Kartenwerkzeug, das die Basisfunktionalität zur Anzeige und Bearbeitung der Karte auf Basis von Eclipse Draw2D realisiert.

Das Werkzeug bietet die sogenannten Layer (Ebenen), die übereinander gelegt werden, um in der Summe die Kartenansicht zu realisieren. Beispiele für Layer sind die Hafenkarte selbst oder der Layer, der Baustellen visualisiert. Daneben sind die Interaktionsformen, mit denen man Elemente auf der Karte direkt manipulieren kann, ein wichtiges Konzept des Kartenwerkzeugs.

Im Sinne eines generischen und erweiterbaren Leitstands bietet das Kartenwerkzeug Eclipse-Extension-Points für Layer und Interaktionsformen an. Neue Plug-ins können das Kartenwerkzeug also um Layer und passende Interaktionsformen ergänzen. Ein Plug-in kann mehrere Layer unterschiedlichen Typs (Hintergrund, Editor, Toolbar) anbieten. Über den Typ werden die Layer der unterschiedlichen Extensions in eine geeignete Reihenfolge gebracht.

Die von den Plug-ins bereitgestellten Interaktionsformen basieren primär auf der Interpretation von Maus- und Touch-Ereignissen. Die Interaktionsformen werden in die Verarbeitung der Obenflächenereignisse mit einbezogen. Die zugrunde liegenden Events, die das UI-Framework SWT von RCP liefert, werden dabei zu höherwertigen Gesten aggregiert. Beispielsweise werden aus den eher primitiven Touch-Ereignissen Click-, Drag- oder Pinch-Gesten abgeleitet (für den mit Touch bedienten Systemteil „Etikettensystem“, siehe Abschnitt „Erweiterungen“).

Das Kartenwerkzeug bringt zum Beispiel Interaktionsformen mit, die die grundlegende Navigation auf der Karte realisieren. Die Baustellenerfassung ergänzt das Editieren von Polygonen und die Interaktion mit erfassten Baustellen.

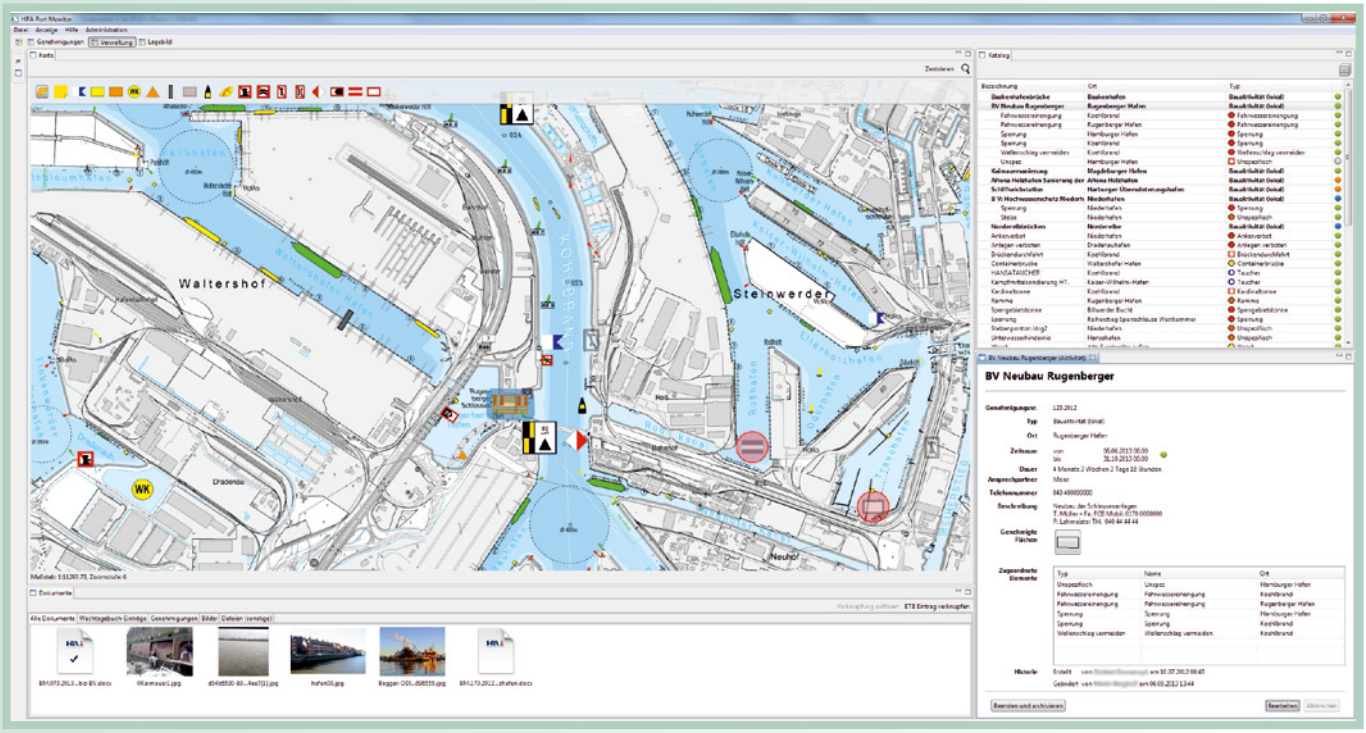

Abb. 3: Verwaltungsperspektive

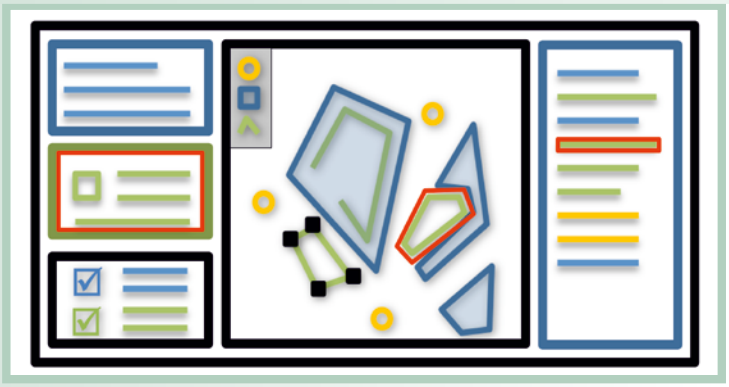

Abb. 4: Zusammenspiel der Werkzeuge (4 Plug-ins, schematisch)

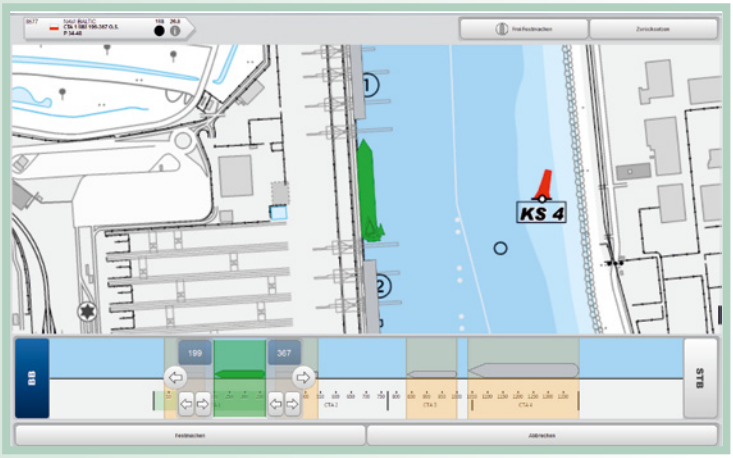

Abb. 5: Touch-basiertes Anlegen von Schiffen mit dem Etikettensystem

Natürlich nutzen der Port Monitor und das Kartenwerkzeug auch die Standard-RCP-Extension-Points, zum Beispiel für Editoren, Viewer, Menüeinträge.

Erweiterungen

Der Integrierte Port Monitor ist seit Einführung schrittweise erweitert worden. Unter anderem sind hinzugekommen:

- Der *Mobile Port Monitor*, eine Tablet-App, die auf den Fahrten im Hafengebiet eingesetzt wird, um Baustellen zu kontrollieren und vor Ort neu entdeckte Hindernisse zu erfassen [BrBe13].
- Das *Etikettensystem*, mit dem die Nautiker die aufkommenden und abgehenden Schiffe als „Etiketten" frei anordnen und für sich organisieren können, und darüber hinaus im System dokumentieren, dass Schiffe an bestimmten Liegeplätzen angelegt haben. Das Etikettensystem ist eine auf Basis von Eclipse RCP implementierte Touch-Anwendung, die unter Windows 8 läuft und die die Anwender an einem waagerecht aufgestellten, Touch-fähigen Monitor bedienen.

Minimal-invasive Integration

Was ist das?

Im Umfeld der Nautischen Zentrale existieren zahlreiche Spezialsysteme, die jeweils bestimmte Aspekte der gesamten Hafensituation verwalteten, zum Beispiel das Geoinformationssystem für die Karten und langlebige Markierungen darauf, das Informationssystem DV Elbe zur Verwaltung unter anderem der Schiffsankünfte und Liegeplatzzuweisungen – und viele weitere.

Bei der Integration dieser Systeme haben wir uns an dem Leitbild der „minimal-invasiven Integration" orientiert, das so beschrieben werden kann:

- Binde die Systeme zuerst lesend ein.
- Belasse die Systeme in ihrer bisherigen Funktion und als primäre Eingabesysteme für diejenigen Daten, für die sie führend sind.

- Wenn im Port Monitor Daten aus den Backend-Systemen verändert werden, speichere die veränderten Daten in diese Systeme und lese sie auch in Zukunft wieder von dort.

Wir verstehen Integration nicht als Einbahnstraße, sondern stellen Services wiederum anderen Systemen zur Verfügung, die sich unsere Aufbereitung oder Verknüpfung der „Rohdaten" zunutze machen können.

Eine Herausforderung im Umfeld von Leitständen ist die Integration auf Client-Ebene. Wir haben viele Leitstände kennengelernt, in denen Bildschirme mit isolierten Anwendungssystemen aneinandergereiht sind. Um darüber hinauszugehen, haben wir den Port Monitor lose mit zentralen Fachanwendungen gekoppelt. Es ist uns gelungen, die Arbeitsabläufe stark zu vereinfachen, indem wir im Port Monitor anbieten, in einen anderen Client zu wechseln und dabei die relevanten Elemente aufzurufen – den Arbeitskontext beim Wechsel also „mitzunehmen".

Services

Backend- und Port-Monitor-eigene Funktionalität werden zu einem großen Teil als Services (SOAP, WSDL-First) zentral bereit gestellt. Servicezugriffe werden je nach Bedarf gecachet. Für die Aktualisierung der Caches nutzen wir den JMS-Rückkanal, wo dies möglich ist, sonst Polling.

Im Gegensatz zum Standard-Ansatz von RCP (bzw. OSGi) depublizieren wir Services auf dem Client nicht, wenn sie nicht verfügbar sind. Dies hätte nämlich den Nachteil, dass der auf Services zugreifende Code damit umgehen müsste, dass diese jederzeit „verschwinden" können. Stattdessen sind unsere Client-seitigen ServiceProxys dafür zuständig, Service-Connections zum Backend zu verwalten. Ist die Backend-Funktionalität nicht verfügbar, ermöglichen sie, wo sinnvoll, weiterhin den lesenden Zugriff gecacheter Daten. Das bedeutet, dass man mit dem System lesend ganz normal interagieren kann, während nur Änderungen nicht möglich sind.

Herausforderungen des Technologiestacks

Die Kombination aus Eclipse RCP mit dem darin enthaltenen OSGi-Laufzeitsystem, Spring und Webservices stellt für die Entwicklung eine Herausforderung mit vielen Fußangeln dar. Hier nur ein Beispiel:

Problem: In einer RCP-Anwendung sind viele Elemente (insbesondere Oberflächenkomponenten) als Extensions zu realisieren. Typischerweise wird der Lebenszyklus eines Extension-Objektes von der Plattform bzw. vom Extension-Point bestimmt. Auf der anderen Seite wollen wir diese Objekte als Spring-Beans definieren, um Dependency Injection (DI) nutzen zu können.

Lösung: Verwendung der „SpringExtensionFactory" [SpringXF]. In der Datei plugin.xml gibt man als „Class" für die Extension die SpringExtensionFactory an. Diese zieht dann eine Bean, deren ID mit der ID der Extension übereinstimmt.

Ausblick

Auch in den kommenden Monaten stehen Erweiterungen des Port Monitors an. Im Frühjahr 2014 soll die neue Nautische Zentrale eröffnet werden, zu der auch eine großformatige Videowall (Videowand, bis zu 15x Full HD) gehören wird. Der Port Monitor ist eine wichtige Bildquelle für diese Videowall und wird darauf speziell angepasst werden.

Weiterhin sind neue mobile Anwendungen im Zusammenhang mit dem Port Monitor in Planung.

Fazit

Wir führen den Erfolg des Port Monitors auf drei wesentliche Faktoren zurück: Die gründliche Arbeitsplatz- und Aufgabenanalyse zu Beginn des Projekts, die Konstruktion einer dafür passenden, flexiblen Leitstandsplattform und den von uns verfolgten, minimal-invasiven Integrationsansatz.

Unsere Projekterfahrungen zeigen, dass die Eclipse Rich Client Platform trotz ihrer Komplexität gut geeignet ist, um damit Systeme zu bauen, die mehrere, unterschiedliche Arbeitsplätze passend unterstützen sollen. Das Zusammenspiel mit Spring hat sich bewährt und unter anderem die Testbarkeit der Anwendungskomponenten erheblich verbessert.

Der Nutzen des Integrierten Port Monitors ist durch den schrittweisen Ausbau und die Einbindung zusätzlicher Informationsquellen überproportional gewachsen; in der nächsten Zeit soll das System darum erneut mehrfach erweitert werden. Die Architektur ist hierfür tragfähig, wie die bisherigen Entwicklungsschritte bewiesen haben.

Somit ist der Port Monitor heute ein integrierter und weiter wachsender Leitstand in Hamburgs Herzkammer: dem Hafen.

Literatur und Links

[AHOI] Forschungsprojekt Arbeitsgerechte Neugestaltung der Nautischen Zentrale des Hamburger Hafens, Uni Hamburg, 2010, `http://uninews-online.de/2010/12/22/arbeitsgerechte-neugestaltung-der-nautischen-zentrale-des-hamburger-hafens/`

[BrBe13] H. Breitling, M. Berghoff, App für den Hamburger Hafen, in: JavaSPEKTRUM, 06/2013

[GeneAL] Forschungsprojekt Generische Architektur für Leitstände, `http://www.c1-wps.de/forschung/geneal.html`

[HPA12] Klarmachen zum Klicken, Seiten 24 bis 27 aus dem Geschäftsbericht 2012 der HPA, `http://www.hamburg-port-authority.de/de/presse/broschueren-und-publikationen/Documents/Geschäftsbericht%202012%20Imageteil.pdf`

[PortMonitorV] Video zum Port Monitor, `http://www.youtube.com/watch?v=9n94arrMAhQ`

[SpringXF] Spring Extension Factory, `https://github.com/martinlippert/spring-extension-factory`

[UTM] Universal Transverse Mercator, `http://de.wikipedia.org/wiki/UTM-Koordinatensystem`

Holger Breitling ist Senior Softwarearchitekt bei der C1 WPS GmbH und Mitglied der Geschäftsleitung. Er beschäftigt sich mit Integrations- und Transformationsarchitekturen und Softwareentwicklungsvorgehen. E-Mail: holger.breitling@c1-wps.de

Arne Scharping ist Softwarearchitekt bei der C1 WPS GmbH. Er beschäftigt sich mit der Konzeption und Weiterentwicklung individueller Anwendungssysteme. In den letzten Jahren bildet das Umfeld der Hafenlogistik einen inhaltlichen Schwerpunkt. E-Mail: arne.scharping@c1-wps.de