

i-Portal-Patterns

Lösungsmuster für wiederkehrende Anforderungen

Johannes Willkomm
Dr. (Ph.D.) Bernhard Humm

sd&m AG
software design & management
Herrnstraße 57
63065 Offenbach am Main
johannes.willkomm@sdm.de
bernhard.humm@sdm.de

Abstract: Ein Portal unter Einsatz automatischer Mechanismen zum Erzeugen von Seiten im Hochlastbereich stellt eine Vielzahl von Anforderungen an Integrator und Betreiber. Das Paper beschreibt die Kernaussagen von Mustern für Integrationsprojekte im Portalbereich (i-Portal-Patterns). Ein konkretes i-Portal-Pattern wird im Detail erläutert.

1 Einleitung

Internet-Portale führen Informationen aus unterschiedlichen Quellen zusammen und stellen sie Portalnutzern über einen zentralen Einstieg zur Verfügung. Beim Design der Technischen Infrastruktur, der Architektur und der Inbetriebnahme von Portalen kann durch die Berücksichtigung einfacher Regeln eine Vielzahl von Hardware eingespart, das System maßgeblich stabiler gemacht oder aber auch eine Inbetriebnahme neuer Software mit geringem Risiko durchgeführt werden. Diese einfachen Regeln zur Integration von Portalen haben wir i-Portal-Patterns getauft.

Sicherlich gibt es bereits eine Vielzahl von aufgestellten Patterns für die unterschiedlichsten Themengebiete. An dieser Stelle seien die Design Patterns [Ga95] genannt. Eine Sammlung von Patterns unterschiedlichster Art findet man unter anderem im Portland Pattern Repository [Pr]. Die i-Portal-Patterns grenzen sich jedoch insofern von diesen Patterns ab, als dass der Fokus das Portalgeschäft ist. Zusätzlich wird das Thema weiter eingegrenzt, um technische Hilfestellung für die Integration von Portalen zu geben.¹

¹ Es sei angemerkt, dass IBM mit ihren *Integration Patterns for e-Business* [Ib] zwar einen ähnlichen Ansatz wählt, das Thema jedoch weiter gefasst ist (EAI) und auch die Informationsstruktur eine andere ist.

Wir haben konkrete Projekterfahrungen zur Integration von Portalen in Form einer sd&m-internen i-Portal-Patternsammlung kondensiert. In diesem Paper stellen wir die Kernaussagen einiger dieser i-Portal-Patterns vor und beschreiben eines im Detail.

2 Kernaussagen der i-Portal-Patterns

i-Portal-Patterns haben wir für die Bereiche Browser, Server und Technische Infrastruktur für den firmeninternen Wissensspeicher auf Grundlage unserer Projekterfahrungen aufgestellt. Diese i-Portal-Patterns folgen einer immer gleichen Struktur, die sich an die Struktur der Design-Patterns [Ga95] anlehnt. Die im Folgenden aufgeführten Kernaussagen sollen einen Einblick geben, welcher Art diese einfachen Regeln sind. Für eine Detaillierung der Aussagen möge sich der geneigte Leser direkt an die Autoren des Papers wenden.

2.1 Kernaussagen der i-Portal-Patterns für den Bereich *Browser*

HTML-Kompatibilität zu Browsern und Betriebssystemen: Die pixel- und farbgenaue Unterstützung vieler Browser-Betriebssystem-Kombinationen für ein Portal ist aufwändig. Daher sollte man sich bezüglich eines perfekt abgestimmten Layouts auf wenige Browser-Betriebssystem-Kombinationen beschränken.

Performance-Optimierung für das Rendering: Das Rendering hat einen maßgeblichen Anteil an der Gesamt-Performance einer Browser-Anwendung. Durch eine Vereinfachung der HTML-Struktur und Reduzierung der Seitengröße kann die Rendering-Zeit erheblich reduziert werden.

Client-seitiges Caching: Für Seiten, welche umfangreiche Daten enthalten, z.B. Bilder, soll ein Caching in Browsern und Proxies durchgeführt werden. Dafür sollen Meta-Informationen im HTTP-Header beziehungsweise HTML-Head verwendet werden. Um unerwünschte Caching-Effekte in Proxies und Browsern zu vermeiden, werden Bilder mit Versions-Nummern eindeutig gekennzeichnet.

2.2 Kernaussagen der i-Portal-Patterns für den Bereich *Server*

Auslieferungs-Mechanismen für HTML-Seiten: Zur Auslieferung von Seiten stehen die Mechanismen (a) Offline-Generierung, (b) Caching und (c) dynamische Generierung zur Verfügung. Der Auslieferungsmechanismus soll anhand der Anzahl der Seiten und der Zugriffs-Häufigkeit pro Seite bestimmt werden.

Fachliches Clustering für horizontale Skalierung: Für eine optimale horizontale Skalierung von Web- und Portal-Servern sollen Web-Seiten in fachlichen Clustern ausgeliefert werden.

Performance-Messung für Portale: Aussagen über ausreichende Performance eines Portals kann nur am Produktivsystem gemacht werden. Daher ist eine gestufte Einführung bei Hochlastportalen sehr zu empfehlen.

2.3 Kernaussage des i-Portal-Patterns für die technische Infrastruktur

Relaunch eines Portals: Der Relaunch eines Portals sollte über die Einführung eines neuen Hostnamens oder die Änderung der Lastverteiler-Konfiguration erfolgen. Bei Umstellung über eine DNS Änderung sollte der alte Web-Server in keinem Fall mehr von außen Verfügbar sein.

3 i-Portal-Pattern für „Auslieferungsmechanismen für HTML-Seiten“

Kernaussage: Zur Auslieferung von Seiten stehen die Mechanismen (a) Offline-Generierung, (b) Caching und (c) dynamische Generierung zur Verfügung. Der Auslieferungsmechanismus soll anhand der Anzahl der Seiten und der Zugriffs-Häufigkeit pro Seite bestimmt werden.

Voraussetzungen: HTML-Seiten werden automatisch erzeugt.

Gegenanzeigen: keine

Problem: Bei der Erstellung eines Portals sind Stabilität, Performanz und Publikationsgeschwindigkeit meist hoch priorisierte Anforderungen, die nicht immer miteinander vereinbar sind. Ferner haben Portale bezüglich ihrer Seiten- und Sitestruktur unterschiedliche Anforderungen, die es zu berücksichtigen gilt.

Wie unterscheiden die Mechanismen (a) Offline-Generierung, (b) Caching und (c) dynamische Generierung zur Auslieferung von Seiten. Es ist der Auslieferungsmechanismus so zu wählen, dass unter minimalem Ressourcen-Einsatz die Anforderungen an eine performante und stabile Site mit hoher Publikationsgeschwindigkeit optimal erfüllt werden.

Lösung: Zur Identifikation der Anforderungen an die Site ist in einem ersten Schritt zu überprüfen, in welche der drei folgenden Kategorien die Site eingeordnet werden kann:

1. **Kategorie 1:** Es müssen wenige unterschiedliche Seiten ausgeliefert werden und es besteht eine hohe Zugriffshäufigkeit auf die einzelnen Seiten. Alle Portalnutzer erhalten die gleichen Seiten. Beispiel: die Startseite der T-Online-Portals www.t-online.de.
2. **Kategorie 2:** Die Seiten sind bedingt Portalnutzer-abhängig. Viele Portalnutzer erhalten identische Seiten. Es gibt wenige Kombinationen zur Zusammenstellung der Inhalte zu Seiten. Beispiel: Nachrichtenseiten unter www.bild.t-online.de.
3. **Kategorie 3:** Die Seiten sind stark Portalnutzer-abhängig. Tendenziell erhält jeder Portalnutzer eine für ihn individuell zusammengestellte Seite. Beispiel: das Telefonbuch im Internet www.teleauskunft.de.

Abbildung 3.1 verdeutlicht die Unterscheidung in die drei Kategorien. In Kategorie 1 werden nur wenige unterschiedliche Seiten ausgeliefert. Die Anzahl der Anfragen auf eine einzelne Seite kann hier sehr hoch sein.

Für Kategorie 3 ist dieses Verhältnis umgekehrt. Eine identische Seite wird nur selten mehrmals angefragt.

Kategorie 2 deckt solche Fälle ab, die sich nicht in die Kategorie 1 oder 3 einordnen lassen.

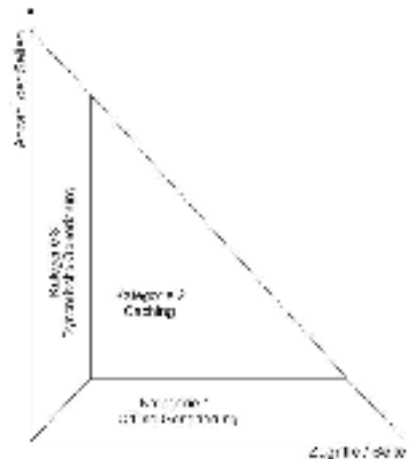


Abb. 3.1

Abhängig von der Kategorie ist dann folgender Auslieferungsmechanismus zu wählen:

Bei Kategorie 1 sollte die *Offline-Generierung* gewählt werden, d.h. die Veränderung einer Seite durch einen Redakteur löst einen Generierungs-Event aus. Eine HTML-Seite wird durch diesen Generierungs-Event statisch einem Webserver zur Verfügung gestellt. Dieser liefert aufgrund eines HTTP-Requests diese Seite aus. Ein HTTP-Request führt nie zu einem Generierungs-Event.

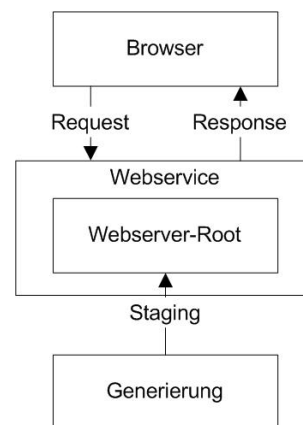


Abb. 3.2

Für Kategorie 2 sollte *Caching* gewählt werden, d.h. nur der erste HTTP-Request für eine Seite wird von einem Generierungs-Event beantwortet. Das Ergebnis wird gleichzeitig im Cache abgelegt. Solange die Seite nicht redaktionell verändert wird, werden alle weiteren HTTP-Requests aus dem Cache (z.B. Web-Server-Root-Verzeichnis) beantwortet. Abbildung 3.3 verdeutlicht diesen Ablauf. Im Falle einer redaktionellen Veränderung durch einen Redakteur, wird die Seite aus dem Webserverroot entfernt und die neuen Inhalte aufgrund eines neuen HTTP-Requests mit neuen Inhalten generiert.



Abb. 3.3

Für Kategorie 3 sollte Dynamische Generierung gewählt werden, d.h. jeder HTTP-Request wird von dem Webserver an einen Generator weitergeleitet und durch das Ergebnis einer individuellen Seiten-Generierung beantwortet.



Abb. 3.4

Oberhalb Diagonalen von Abbildung 3.1 beginnt ein Bereich, in dem man an die Grenzen der technischen Möglichkeiten stößt.

Wir kennen keine konkreten Schwellwerte für die Parameter *Anzahl der Seiten* und *Anzahl der Zugriffe pro Seite*. Die Schwellwerte verschieben sich abhängig von der Priorisierung von Publikationsgeschwindigkeit und Online-Performance. Ist die Online-Performance wichtiger als die Publikationsgeschwindigkeit, so verschiebt sich das Spektrum zu Gunsten der Offline-Generierung – umgekehrt in Richtung dynamischer Generierung.

Unter Umständen muss auch innerhalb eines Portals zwischen unterschiedlichen Auslieferungsmechanismen unterschieden werden, wenn die Anzahl der Hits pro Seite bezogen auf die einzelnen Seiten stark variieren.

Beispiel:

Aufgrund der hohen Lastanforderungen waren Optimierungsmaßnahmen für die T-Online-Themenportale, z.B. t-news.t-online.de, notwendig. Basis für diese Optimierungsmaßnahmen war das vorgestellte i-Portal-Pattern. Mit empirischen Mitteln wurde über alle Portale hinweg ermittelt, welche Seiten am häufigsten angefragt wurden und wie viele dieser Seiten mittels Offline-Generierung mit einer akzeptablen Publikationsgeschwindigkeit generiert werden können. Hiermit konnte eine sinnvolle Menge von Seiten definiert werden, die über die Offline-Generierung ausgeliefert werden. Der Rest der Seiten wird über einen Caching-Mechanismus ausgeliefert. Das Ergebnis war, dass ein Drittel der Hardware, die zuvor benötigt wurde, um alle Seiten mittels Caching auszuliefern, eingespart werden konnte.

4. Zusammenfassung

Bei Portal-Integrationsprojekten wird der Integrator auf vielen Ebenen, die über das eigentliche Software-Design hinausgehen, herausgefordert. Für diese Ebenen stellen i-Portal-Patterns eine Hilfe dar, die den Integrationsprozess beschleunigen oder mit denen man mit einfachen Mitteln komplexe Herausforderungen meistern kann. Konkret haben wir dies am Beispiel des Patterns „Auslieferungsmechanismen für HTML-Seiten“ gezeigt, durch dessen Anwendung bei T-Online ungefähr ein Drittel der zuvor benötigten - Hardware (Portal-Server) eingespart werden konnte. Auch die Anwendung des Patterns „Fachliches Clustering für horizontale Skalierung“ hat eine deutliche Reduktion der benötigten Hardware gebracht. Durch die Anwendung des Patterns „Relaunch eines Portals“ konnte beim Internet-Auftritt der Bild-Zeitung die Einführung einer neuen Portal-Software für den Portalnutzer reibungslos durchgeführt werden.

Es gilt nun, weitere Patterns im Rahmen unserer Tätigkeit als Portal-Integrator aufzustellen und die bestehenden Patterns zu verfeinern.

Literaturverzeichnis

- [Ga95] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns, Elements of Reusable Objectoriented Software, Addison-Wesley 1995.
- [Ib] IBM-Patterns for e-Business
<http://www-106.ibm.com/developerworks/patterns/index.HTML>
- [Pp] Portland Pattern Repository
<http://c2.com/ppr/>
- [Si02] Johannes Siedersleben (Hrsg.): Quasar: Die sd&m Standardarchitektur, Teil 1, sd&m-Research 2002.