

Abbildung 6-4: Prozessfähige SOA

Man beachte aber, dass BPEL, als Web-Services-Standard für die Komposition und Orchestrierung von Services, keinerlei Möglichkeit für die direkte Interaktion mit Anwendern über entsprechende User-Interfaces anbietet. Dies hat wichtige Auswirkungen auf das generelle Design von SOA-Landschaften, die in Kapitel 9 diskutiert werden.

6.4.2 Service-Zustand versus Backend-Zustand

Beim Umgang mit Prozess-Services muss man eine wichtige Designentscheidung treffen: Soll der Zustand eines Geschäftsprozesses im Service oder in einem Backend verwaltet werden? Die Antwort hat damit zu tun, wie relevant der Zustand aus fachlicher Sicht ist:

- Wenn der Zustand des Prozesses »juristisch relevant« ist, ist es vermutlich angemessener, den Zustand in einem herkömmlichen Backend zu verwalten.
- Wenn der Zustand des Prozesses noch keine Relevanz für das Business hat, sind zustandsbehaftete Services eher angebracht.

Ein typisches Zeichen für juristische Relevanz ist das Vorhandensein einer fachlichen ID für diesen Geschäftsprozess (Bestellnummer, Buchungs-Code usw.), auf die man sich dann Tage, Wochen oder sogar Jahre später noch beziehen kann. Wenn man als Service-Anbieter auch später noch auskunftsfähig über einen laufenden Geschäftsprozess sein muss, sollten die Prozessdaten in einem Backend liegen.

(siehe Abschnitt 15.1.2 auf Seite 238 für eine Diskussion von zustandsbehafteten Services).

Ein typisches Beispiel für einen Prozess-Services ist ein Service für einen Warenkorb. Sein Zustand enthält die jeweils über einzelne Service-Aufrufe hinzugefügten Waren (Produkte oder Dienstleistungen), möglicherweise ergänzt um die Kundendaten. Der Prozess endet in der Regel damit, dass aus dem Warenkorb eine Bestellung wird.

Da Prozess-Services typischerweise zustandsbehaftet sind, müssen in einer SOA-Landschaft einige Mechanismen vorgesehen werden, um den Zustand über mehrere Service-Aufrufe verwalten und dabei immer noch linear skalieren zu können (siehe Abschnitt 15.1.3 auf Seite 241). Dazu kann gehören, dass man sogar den Client bzw. das Frontend wechseln können muss, was bedeutet, dass zu der logischen Session mehrere physikalische Sessions in den Frontends gehören.

Ein typisches Beispiel dafür wäre ein Service zum Abschluss einer Versicherung. Der Kunde mag den Prozess online über ein Webportal einer Versicherung oder eines Vermittlers starten. Im Laufe des Prozesses werden aber andere Frontends benötigt (Call-Center oder Back-Office-Arbeitsplätze), die die bisher angefallenen Daten abfragen und um neue Daten ergänzen. Der Prozess wäre erst durch Inkrafttreten der Versicherung oder deren Nichtzustandekommen beendet.

Bei derartigen Geschäftsprozessen muss auch ein »Failover« berücksichtigt werden. Darunter versteht man die Fähigkeit, dass der Zustand des Prozesses erhalten bleibt, auch wenn das System, das den aktuellen Zustand verwaltet, ausfällt (ein anderes System springt dann sozusagen in die Bresche).

Anforderungen wie zustandsbehaftet und Failover sind der Hauptgrund, dass für diese Services eine eigene Kategorie eingeführt wird. Hinzu kommt aber, dass sich langlaufende Services auch in fachlicher Hinsicht (vor allem in Bezug auf nichtfunktionale Anforderungen) von kurzlaufenden Composed-Services unterscheiden. Erneut muss hier in der konkreten Umsetzung entschieden werden, ob der Unterschied wirklich so relevant ist, dass sich eine eigene Kategorie lohnt. [KrafigBankeSlama04] weist darauf hin, dass Composed- und Prozess-Services als eine Kategorie betrachtet werden können, und [Erl05] verwendet den Begriff »Prozess-Services« sogar grundsätzlich für diese beiden Arten von Service zusammen.

6.4.1 Prozessfähige SOA

Die Einführung von Prozess-Services führt zu einer Ausbaustufe, die »prozessfähige SOA« genannt wird (auch dieser Begriff wurde von [KrafigBankeSlama04] übernommen). Wie vorher erwähnt, ermöglicht die zusätzliche Schicht der Prozess-Services, Geschäftsprozesse zu verwalten, die über mehrere Frontends laufen und auch durch menschliche Interaktion unterbrochen werden können (siehe Abbildung 6-4).

tationen führen. Der Standard definiert nur ein einheitliches Format für die inhaltlichen Schritte, die in einem Prozess/Service durchgeführt werden, sodass sich das Verhalten beim Wechsel von einem zu einem anderen Tool nicht ändern sollte.

Der genaue Name von BPEL hat sich mehrfach geändert. Initial hieß es BPEL4WS, bevor es, um den allgemeinen Web-Services-Konventionen zu genügen, in WS-BPEL umbenannt wurde. Es handelt sich um einen Web-Services-Standard, weil der resultierende Prozess technisch ein Web-Service ist, der als solcher weiterverwendet werden kann. Semantisch kann es sich um einen Composed- oder Prozess-Service handeln. Ein Nutzer kann den Prozess also einfach als Web-Service aufrufen (oder als solchen weiter orchestrieren).

Handelt es sich bei BPEL also um einen reinen Web-Services-Standard? Die Antwort lautet nein. In den BPEL-Dateien gibt es zwei verschiedene Bereiche. In einem Bereich werden die Struktur und das Verhalten der Prozesse/Services beschrieben. Im anderen Bereich kann dann zu den aufgerufenen Services ein Binding angegeben werden, das prinzipiell beliebige Technologien nutzen kann. Ein Binding zu anderen Web-Services ist dabei standardisiert, aber es ist in der Praxis nicht schwierig, auch Services aufzurufen, die mit anderen Technologien bereitgestellt werden. Neben verschiedener Middleware können prinzipiell auch native Technologien wie J2EE-Methoden oder Datenbankhersteller verwendet werden (es überrascht sicherlich nicht, dass ein Datenbankhersteller wie Oracle in seiner BPEL-Implementierung eine direkte Unterstützung für den Aufruf von Stored-Procedures und SQL-Statements als »Services« anbietet). Hier muss aber natürlich geklärt werden, wie portabel derartige Bindings sind.

7.4.3 Weitere Standards zur Geschäftsprozessmodellierung

BPEL ist nicht der einzige zur Verfügung stehende Standard zur Modellierung von Geschäftsprozessen. Abbildung 7-7 liefert dazu einen Überblick.²

In diesem Diagramm kann man zwischen den folgenden Hauptzweigen bei der Evolution von Standards für Geschäftsprozessmodellierung unterscheiden:

- Der derzeit bekannteste Zweig ist BPEL, der bereits im vorherigen Unterabschnitt diskutiert wurde.
- Ein weiterer wichtiger Zweig begann mit der »Workflow Management Coalition« (WfMC), die 1993 gegründet wurde, um Standards für Workflow-Management-Systeme zu verabschieden. Der erste Standard war die »Workflow Process Definition Language« (WPDL), woraus unter dem Einfluss von XML die »XML Process Definition Language« (XPDL) hervorging (siehe [WfMC] für Details).

²Ein herzlicher Dank geht an Dr. Martin Bartonitz, der in zwei Artikeln die Basis für dieses Diagramm legte und freundlicherweise die Erlaubnis gab, diese Diagramme zu verwenden (siehe [Bartonitz05] und [Bartonitz06]).

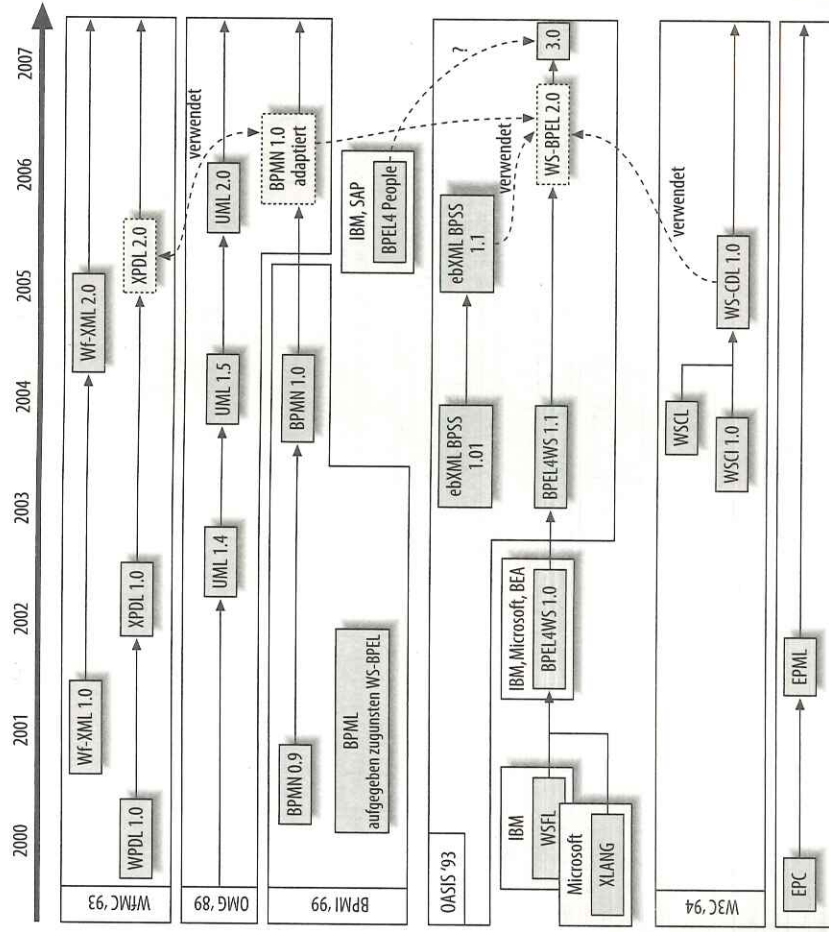


Abbildung 7-7: Standards zur Geschäftsprozessmodellierung

- Der dritte wichtige Standard ist die »Business Process Modeling Notation« (BPMN), die initial von der »Business Process Management Initiation« entwickelt wurde, inzwischen aber von der »Object Management Group« (OMG) gepflegt wird (siehe [BPMN] für Details).

Folgende Standards spielen ebenfalls eine Rolle:

- WfXML, ein Standard der WfMC, der definiert, wie eine Prozessdefinition (formuliert in BPEL oder XPDL) in einer Prozess-Engine ausgerollt (deployt) wird.
- UML (die Unified Modeling Language), die gleich mehrere grafische Notationen bietet, um fachliche Abläufe zu definieren. Dazu gehören Use-Cases, Sequenzdiagramme und vor allem Aktivitätsdiagramme.
- WS-CDL (die Choreography Definition Language), ein Standard zur Definitionen von Geschäftsprozessen mit Hilfe von Choreografie (siehe Abschnitt 7.6 auf Seite 121).