6 Message-Queues und Message-Server

Der wesentliche Unterschied zwischen moderner objektorientierter Programmierung und traditioneller imperativer Entwicklung liegt in der gezielten Verwendung von Nachrichten. Messages erlauben ein gleichberechtigtes Nebeneinander von spezialisierten Programmkomponenten, die sich gegenseitig über Status und Ergebnisse informieren und auf die ausgelösten Events reagieren.

Eine Message-Queue ist intelligente Software, deren Aufgabe es ist, den Austausch von Informationen zwischen Programmen sicherzustellen. Dabei handelt es sich nicht um eine neue Technologie, sondern um die Renaissance einer Methodik, die bei Mainframes schon immer gang und gäbe war. Dort liefen Prozesse in Hintergrund-Jobs und Terminals haben grundsätzlich durch Senden von Nachrichtenströmen mit dem Mainframe kommuniziert.

6.1 Anwendungen von Message Queues

6.1.1 Modelle von Message-Queues

Der Gedanke hinter der Verarbeitung von Messages ist weithin ein Spiegelbild der Kommunikationsabläufe im realen Leben. Jemand (der Client) benötigt die Hilfe eines kompetenten Dienstleisters (des Servers). Häufig kennt der Client aber niemanden, der die Dienstleistung erbringen kann.

Stellen wir uns vor, der Client sei ein Handwerksmeister und er suche einen Betrieb, der ihm seine Werkstatt renoviere und neu einrichte. Natürlich möchte er das beste Angebot haben. Er hat nun mehrere Möglichkeiten:

- **Branchenverzeichnis – Directory Service**
  Er kann entweder in einem Verzeichnis wie den Gelben Seiten oder »Wer liefert was?«, Google oder Yahoo! nachsehen und dann alle Einrichter anschreiben und ein Angebot einholen. Das Problem entsteht schon beim Durchsuchen des Verzeichnisses: Es gibt mehr als eine Branche, die Werkstätten einrichtet, darunter auch solche, die dies gar nicht ausdrücklich erwähnen, sondern die Leistung unter anderem Namen erbringen.

Directory Service

Im Falle einer Message-Queue nennt man eine solche Dienstleistung einen *Directory Service*. Da der Anfragende dabei in der Regel auf eine Antwort des Directory Service wartet, spricht man hier von einer *pseudo-synchronen Verarbeitung*.

- **Ausschreibung und Kleinanzeige**
  Der Handwerksmeister kann aber auch in einer Zeitung oder im Internet die Leistung ausschreiben und auf eingehende Angebote warten. Leider steht und fällt diese Idee mit der Reichweite der Anzeige, also damit, wie viele geeignete Bewerber die Anzeige überhaupt sehen.

Broadcast and Subscribe

Im Falle einer Message-Queue nennt man dies *Broadcast and Subscribe*, damit ist gemeint, dass eine Nachricht verteilt wird und man anschließend auf eine Antwort wartet. Alternative Namen dafür sind auch *Fire and Subscribe* und *Mail-Out*.

- **Spezialdienst oder Nachrichtendienst**
  Unser Meister hat aber noch eine bessere Idee, er möchte jemanden fragen, der vielleicht einen geeigneten Betrieb empfehlen kann. Also geht er zum Bürgermeister, der ja alle Handwerksbetriebe vor Ort kennt, und erkundigt sich bei ihm, wer denn genau hierfür geeignet sei. Alternativ kann er auch auf eine Preisagentur zurückgreifen, demnach ein Unternehmen, das sich darauf spezialisiert hat, Anbieter und Dienstleister zusammenzuführen.

Content Driven Delivery

Dies nennt man im Workflow-Umfeld *Content Driven Delivery* und meint die Fähigkeit eines Message-Servers, den Empfänger und die Folgehandlung einer Nachricht aus dem Inhalt abzuleiten. Ein vordefiniertes Prozedere für den Workflow nennt man auch *Standard Operation Procedure* (SOP).

6.1.2 Nutzen der Message-Queues

Für die Verteilung der Nachrichten und ihre Zustellung an eine Applikation gibt es verschiedene Strategien, die von den Message-Queues erfüllt werden. Insbesondere erfüllt die Message-Queue die Routineaufgaben, die ansonsten von jeder Applikation selbst realisiert werden müssten:

- **Guaranteed Delivery**

Sichere Zustellung

Die Angst jedes Senders einer Nachricht ist es, dass die Nachricht nicht zugestellt werden kann. Dabei kann die Nachricht einfach verloren gehen oder sie wird an den Sender zurückgeliefert. In jedem Fall hat der Sender die Pflicht, Maßnahmen zu ergreifen, dass die Daten beim Empfänger ankommen. Da kann es schon einmal vorkommen, dass die

schöne EDIFACT-Nachricht plötzlich auf Papier ausgedruckt via Fax nachgereicht wird.

Die Message-Queue implementiert nun Standardalgorithmen, die versuchen, die gängigen Fälle von Zustellfehlern automatisch und unbemerkt von den Nutzern zu beheben. Bewährte Strategien sind:

- Regelmäßiges Neusenden in Intervallen
- Zustellungen an eine alternative Notfalladresse
- Alarmierung des Bereitschaftsdienstes

- **Auswahl eines Empfängers einer Nachricht**

  Zustellung nach Inhalt, nicht nach Adresse

  Häufig ist dem Sender einer Nachricht überhaupt nicht bekannt, welcher der möglichen Server auf der Empfängerseite der geeignete Partner für die Nachricht ist. Senden wir zum Beispiel einen Werkstattauftrag als E-Mail an »Herrn Müller«. Wenn nun Herr Müller in Pension geht, müsste in allen Client-Systemen die Adresse von Herrn Müller durch die des neuen Zuständigen ausgetauscht werden. Anstatt nun Empfängerlisten auf jedem Client-System zu replizieren und zu pflegen, wird eine logische Adresse, zum Beispiel »Werkstatt«, auf Dauer vergeben. Ändert sich dann die Adresse, wird diese nur auf dem Message-Queue-Server geändert.

  Anwendung für inhaltsbezogene Adressierung

  In der Praxis sendet deshalb der Requester seine Anfrage an den Message-Server mit einer hinreichenden Beschreibung des gewünschten Kommunikationspartners. Typische Anwendungen für eine solche inhaltsbezogene Adressierung sind:

  - Load-Balancing
    Wenn eine Instanz eines Servers – zum Beispiel SAP R/3 – auf mehreren Applikationsservern läuft, werden Anfragen für eine neue Verbindung durch den Message-Server auf den Applikationsserver geleitet, der momentan am wenigsten ausgelastet ist. Die Intelligenz für die Bestimmung der geeigneten Maschine ist dabei in der Message-Queue programmiert.
  - Connection-Pooling
    Bei teuren Leitungsverbindungen zwischen Sender und Empfänger kann es sinnvoll sein, dass die Verbindung nur einmal aufgebaut und von mehreren Nutzern verwendet wird. Als teure Verbindung sind alle Wählverbindungen anzusehen, aber auch Flaschenhälse im Netzwerk, wenn der Verhandlungsaufwand zwischen den Servern unverhältnismäßig hoch ist.

- **EDI-Nachrichten**
  EDI-Nachrichten beschreiben typischerweise im Kopfsatz den Inhalt der gesendeten Nachricht. Es ist Aufgabe des EDI-Message-Servers, die Nachricht gegebenenfalls zu konvertieren und an ein geeignetes Verarbeitungsprogramm weiterzuleiten.

6.2 Message-Server Software

Entsprechend der herausragenden Bedeutung von Message-Queues gibt es mittlerweile auch unzählige Produkte von namhaften und auch von weniger bekannten Herstellern:

- **IBM**
  IBM positioniert sich gleich mit mehreren konkurrierenden Produkten auf dem Markt:
  - IBM WebSphere/MQ und WebSphere Integrator
    Das Produkt hieß bis vor kurzem noch *IBM MQ Series* und ist ein plattformunabhängiger, äußerst robuster Message-Server. Wenn es darum geht, zuverlässig, schnell und ausfallsicher Massen von Nachrichten zu verarbeiten, ist WebSphere/MQ das Produkt der ersten Wahl und wird es wohl auch noch auf lange Sicht bleiben, auch wenn das Einrichten und Programmieren für WebSphere/MQ allemal noch ein Albtraum sein kann.
  - Lotus Notes
    Lotus Notes war eines der ersten Message-Queue-Systeme, die nicht auf Mainframes abliefen. Damit hatte Lotus zwar frühzeitig erkannt, welche Produkte der Markt benötigt, aber ein ungeschicktes Marketing führte dazu, dass zwar zunächst in allen bedeutenden Unternehmen Notes installiert wurde, aber die Nachricht, wozu Lotus Notes geeignet ist, blieb weitestgehend verborgen. Grundsätzlich kann man sagen, wenn Ihr Unternehmen bereits Lotus Notes installiert hat – und die Chancen dafür sind groß –, dann bietet sich diese Installation auch als Middleware für die komplexen Anwendungen an.
  - IBM CrossWorlds
    IBM hat sich mit CrossWorlds noch ein drittes Middleware-Produkt in sein Portfolio gelegt. Die Motivation war vor allem, das SAP-Know-how von CrossWorlds abzugreifen, da es einfacher erschien, zusätzlich zu MQ noch Crossworlds zu installieren, als die Adapter von MQ neu zu schreiben.

- **IONA**
  Bei IONA Orbix XML-Bus geht der Hersteller einen etwas anderen Weg. Im Kern haben wir es weiterhin mit einer Message-Queue zu tun. Der Unterschied liegt in einer vereinfachten Betrachtung des Austauschs der Nachrichten und dem konsequenten Einsatz von XML für den gesamten Workflow.

  Das ist etwa vergleichbar mit der philosophischen Frage, ob sich die Erde um die Sonne oder die Sonne um die Erde drehe. Die Antwort liegt in der Wahl des Modells. Natürlich kann man die Erde in das Zentrum des Sonnensystems stellen und kommt zu den gleichen Erkenntnissen wie mit der Sonne als Dreh- und Angelpunkt. Der Unterschied liegt darin, dass mit einem geozentrischen Weltbild die astronomischen Berechnungen ungleich komplizierter werden. So ist es auch bei der Wahl des richtigen Modells für Message-Queues, und hier hat IONA deutliche Vorteile. IONA gehört neuerdings auch zum Einflussbereich von IBM.

- **Microsoft MSMQ**
  Microsoft bietet für seine NT-Server-Reihen einschließlich Windows 2000 (NT 5) und Windows XP (= NT 6) die Microsoft-Message-Queue, kurz MSMQ, an und bleibt dabei auch seiner bewährten Marketingstrategie treu. Anstatt auf interessierte Kunden zu warten, werden alle Windows-Server-Nutzer mit MSMQ zwangsweise beglückt: Das Programm ist kostenloser Bestandteil jeder NT-Server-Installation. Die Folge ist, dass MSMQ die zahlenmäßig am weitesten verbreitete Message-Server-Software ist, wohingegen die Zahl der Produktivnutzer von MSMQ gegenwärtig noch überschaubar erscheint. Jedoch ist dabei auch zu beachten, dass Microsoft diese Dienste auch für andere Zwecke, etwa für Outlook und Workflow, nutzt.

- **Microsoft BIZTALK**
  Auch Microsoft geht mit mehreren alternativen Produkten in die Arena. Aufbauend auf MSMQ, bietet Microsoft sein professionelles Workflow-Management-System BIZTALK an.

- **Mercator**
  Mercator ist eigentlich ein EDI-Konverter, der aber mittlerweile zu einem flexiblen Message-Handling-System gewachsen ist. Die Basis des Erfolgs von Mercator ist dessen weitgehende und einfache Unterstützung der R/3-Anbindung.

- **Tibco**
  Tibco gehört ebenfalls zu den ganz großen und namhaften Lieferanten von Messaging-Software. Auch Tibco unterstützt SAP R/3 sehr gut, was auch wichtig ist, denn der Austausch von Messages von und nach SAP R/3 ist weiterhin einer der wichtigsten Anwendungen für Middleware.
- **SAP Exchange Infrastructure**
  SAP spielt mit SAP Exchange Infrastructure seit Mitte 2002 ebenfalls in der Liga der Message-Server. Als Bestandteil von SAP-NetWeaver soll SAP XI mittelfristig den gesamten Nachrichtenfluss von, nach und zwischen SAP-Systemen innerhalb einer IT-Landschaft gewährleisten. Auch wenn das Produkt gegenwärtig noch kaum auf dem Markt vertreten ist, verspricht es doch, eines der stärksten mySAP-Produkte zu werden.

6.3 Kommunikation mit Messages-Queues

Orchestrierung aller Nachrichtenströme

Eine Message-Queue ist technisch gesehen eine zentral zugängliche Datenbank, an die ein Client seine Nachrichten sendet. Die Nachrichten werden in der Datenbank zwischengespeichert, bis sie von einer Applikation verarbeitet werden können.

Grundlage für asynchrone Prozesse

Message-Queues erlauben die asynchrone Verarbeitung von Nachrichten und Anfragen, bei der eine Applikation eine Anfrage an die Message-Queue zur weiteren Verfolgung abgibt. Die verarbeitende Applikation meldet das Ergebnis der Verarbeitung dann an die Message-Queue weiter, von der der ursprüngliche Requester die Antwort dann abholen kann, sobald er dafür Zeit hat.

Wenn eine Applikation eine ressourcenintensive Anfrage startet, sendet sie die Anfrage an die Message-Queue und erhält von dieser eine eindeutige Ticketnummer. Mit dieser Nummer kann dann der Vorgang später eindeutig verfolgt werden.

Serialisierung von Anfragen

Die eingegangenen Nachrichten werden von der Message-Queue automatisch in einer sinnvollen Folge serialisiert und an die Applikation weitergeleitet. Abhängig von der gewünschten Aktion speichert die Message-Queue auch Zwischenergebnisse und den Status der Verarbeitung ab.

Es gibt nun zwei Typen von Applikationen: solche, die von der Message-Queue aufgerufen werden, und solche, die regelmäßig die Message-Queue nach neuer Arbeit durchsuchen. Letztere nennt man einen *Daemon*, es sind Programme, die als Hintergrundprozesse permanent ausgeführt werden und nur in Aktion treten, wenn ein bestimmtes Ereignis, wie etwa der Eingang einer Nachricht, auftritt.

Trigger

Alternativ bietet eine Message-Queue immer die Möglichkeit, eine Applikation bei Bedarf zu starten. Das sorgt ebenfalls dafür, dass wertvoller Hauptspeicher nicht durch schlafende Prozesse ausgelastet wird. Zum Beispiel stellt Microsoft in der MSMQ die Applikation *Microsoft MSMQ Trigger* zur Verfügung, die es erlaubt, Nachrichten nach Inhalten, etwa bestimmten Strings, zu durchsuchen und dann eine Aktion auszulösen. Der regelbasierte Mail-Organizer Microsoft Outlook benutzt diese Technik.

Der Daemon meldet die Ergebnisse der Verarbeitung an die Message-Queue zur Abspeicherung zurück. Heutzutage werden die Ergebnisse fast immer als XML-Dokumente ausgetauscht.

Denken wir an die bekannte Situation, eine Datenbank über eine hoch frequentierte Webseite abzufragen. Wenn viele Anfragen gleichzeitig auf die Datenbank zugreifen, können die Wartezeiten unerträglich sein. Mit einer Message-Queue dahinter können Sie die Anfrage in den Hintergrund verlagern und das Ergebnis zu einem späteren Zeitpunkt abholen.

Tatsächlich spielen mächtige Message-Queue-Systeme eine wichtige Rolle im modernen Enterprise-Computing. Wenn die Serverfarmen nur hinreichend groß werden, geht ohne die Message-Queues nichts mehr. Zusammen mit dem Transaktionsmanagement orchestrieren sie die Last der Anfragen (*Load Balancing*)und versuchen durch gemeinsame Nutzung bereits bestehender und laufender Prozesse (*Pooling*), besonders die teuren Ressourcen besser auszunutzen.

Message-Queues steuern auch die Batch-Verarbeitung

Eine weitere Aufgabe einer Message-Queue ist es, Nachrichten für eine verzögerte Verarbeitung aufzubewahren. Dazu gehören vor allem die Massen-Batch-Läufe, etwa das Ausdrucken von Fakturen und Auftragsbestätigungen während der weniger belasteten Nachtzeiten. Auch SAP R/3 bedient sich dieser Mechanismen mit den Einträgen in der Tabelle NAST (für Nachrichtensteuerung), bei der es sich ebenfalls um eine Message-Queue handelt.

Workflow

Baut man nun ganze Ketten von Nachrichten, die Applikationen antriggern und dann ihrerseits wieder Nachrichten versenden, kommen wir zum Prinzip des *Workflows*, heute oft auch *Webflow* genannt, um anzuzeigen, dass die Nachrichten auf Wunsch auch über das Web ausgetauscht werden können.

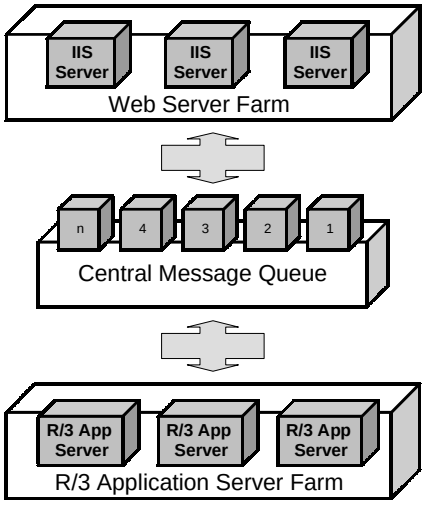

Abbildung 6.1 Message-Queue als Dispatcher in einer Application Server Farm