

IOP – Internet Operating Platform

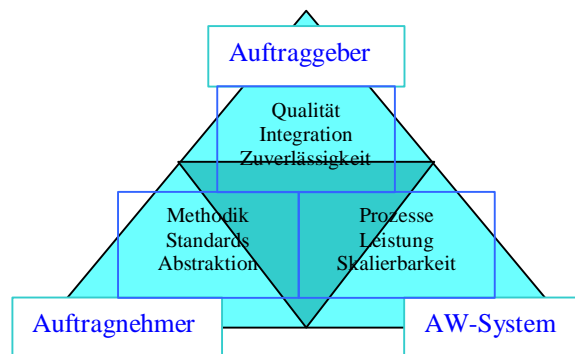
Kurzübersicht, 2002, CronideSoft AG, www.cronidesoft.com

Die CronideSoft AG entwarf das Produkt IOP (Internet Operating Platform) mit dem Ziel, die Entwicklung großer und komplexer Software basierend auf Internet-Technologien zu verbessern und zu optimieren. Die folgenden Abschnitte geben hierzu einen Überblick über die zugrundeliegenden Anforderungen und die daraus resultierenden Lösungen und Konzepte von IOP.

Anforderungen Internet-basierter Anwendungssysteme

In Software-Projekten sind vielfältige Anforderungen aus unterschiedlichen Sichten zu erfüllen oder in Einklang zu bringen, etwa aus Sicht des zu entwickelnden Systems, der Auftraggeber und der Auftragnehmer.

Komplexe Anwendungssysteme, die auf Internet-Technologien und –Architekturen basieren, sind mindestens charakterisiert durch komplexe Geschäftsprozesse und höchste Anforderungen an Leistung und Skalierbarkeit. Vorgefertigte Anwendungssysteme reichen zur Umsetzung solcher Anforderungen nicht aus und müssen durch aufwendige Anpassungen sowie Ergänzungen für den jeweiligen Einsatz geformt und erweitert werden.



Auftraggeber stellen insbesondere sehr hohe Anforderungen an eine hohe Qualität im Ergebnis, an einfache Integration in bestehende Software- und Systemlandschaften, an die Zuverlässigkeit von Termin/Kostenaussagen und an das Potential zur Kostenreduzierung.

Aus Sicht der **Auftragnehmer** entstehen weitere Anforderungen, wie etwa die Planbarkeit von Aufgaben und Teilaufgaben, der aspektbezogene Einsatz von Ressourcen, die Einsetzbarkeit standardisierter Entwicklungsprozesse und Werkzeuge, die Verwendung erprobter Architekturen, die Abstraktion von projektspezifischen / vorgefertigten / bestehenden Anwendungsteilen und die Konzentration auf Geschäftsprozesse anstelle von Technik.

IOP – Internet Operating Platform

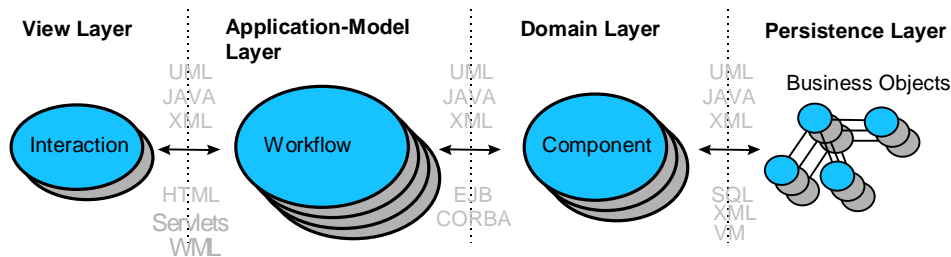
IOP ist eine Entwicklungsplattform, die eine skalierbare Infrastruktur zur Realisierung von eBusiness-Produkten und -Anwendungen bereitstellt:

- IOP ist ein objektorientiertes Java-Framework basierend auf Abstraktionen aktueller Technologien wie Komponentenarchitekturen (EJB, CORBA, eigene), Kommunikationstechniken (HTTP, WAP, Message Queuing usw.) und Frontend-Lösungen (HTML, WML, XML, Servlets, Java Applets usw.).
- IOP bietet ein Höchstmass an Wiederverwendbarkeit bei hoher Flexibilität. Durch seine fraktale Architektur werden Kernrealisierungsmuster und -lösungen auf allen Architekturebenen eingesetzt. Die Realisierung von IOP selbst folgt diesem Prinzip – es ist selbstreproduzierend.
- Sogenannte Mini-Frameworks stellen als Satelliten des IOP-Kerns weitere Infrastruktur für semantisch höherwertige Bereiche (Persistenz, Workflow, Nutzerinteraktion) bereit.
- IOP bietet einen Satz von Richtlinien und Werkzeugen zur Unterstützung aller Phasen des Software-Entwicklungsprozesses.
- IOP ist eine Entwicklungsplattform für Produktentwicklung und Projekte.

IOP Architektur

Die Abbildung beschreibt die wesentlichen Architekturebenen von IOP. Auf allen Ebenen werden durchweg die Kerntechnologien UML zur Modellierung, Java zur Programmierung und XML (oberhalb der Pfeile) zur Konfigurierung/Datenaustausch eingesetzt.

- Die Interaction-Ebene definiert den Anschluss des Anwendungssystems nach außen hin zu Benutzerschnittstellen, ob in Browsern, in stand-alone Anwendungen oder in Mobilgeräten.
- Die Workflow-Ebene definiert Geschäftsprozesse über kombinierbare Aktivitäten. Workflows nehmen Eingaben von der Interaction-Ebene entgegen und liefert Ergebnisse an die Interaction-Ebene zurück. Workflows koordinieren dazu Aufrufe an verteilte Komponenten zur Umsetzung der Geschäftsprozesse.



- Die Component-Ebene steht für wiederverwendbare Teilsysteme wie etwa eine Nutzerverwaltung oder eine Produktverwaltung. Über Components werden auch Fremdsysteme integriert – eine Component liefert in IOP dann die aufrufbare Schnittstelle zum Fremdsystem.
- Die Business-Object-Ebene repräsentiert Geschäftsobjekte, die über Components verwaltet werden, also etwa Personendaten, Produktinformationen usw. Während Components die Business Objects verwalten, stellen Business Objects global eindeutig identifizierbare Informationseinheiten dar. Diese können auf Component- und Workflow-Ebene sowohl mengenorientiert als auch navigierend manipuliert.

Kundennutzen

IOP bietet mit hochgradiger Parallelisierung/Lastverteilung, transparentem Caching und deskriptiver Anfrageergebnisauflösung Konzepte für **effiziente** und **skalierbare** Realisierungen. Diverse Frontend-Ansätze unterstützen überdies verschiedene Bandbreiten (HTML/HTTP versus Java Applets/Applications/RMI) für den Anwenderzugang. Die Architektur ist nachrichtenbasiert, unterstützt verteilte Komponentenmodelle und erlaubt dynamische Konfigurierung von Laufzeitparametern.

IOP bietet einen ausgeklügelten Mechanismus, um alle in seiner Architektur eingesetzten Technologien per Konfiguration austauschen zu können und bietet damit hohe **Flexibilität**. Alle Daten und Inhalte eines Anwendungssystems können zudem logisch und physisch partitioniert und dedizierten Komponenten zugeordnet werden. Für jede physische Komponente stehen dabei unterschiedliche Datenhaltungssysteme im Hauptspeicher, in XML-Dateien oder in Oracle8i zur Ablage und Suche und Verwaltung bereit. Die **Verwaltbarkeit** wird damit beträchtlich verbessert, insbesondere bei technologischen Veränderungen, funktionalen Erweiterungen und auch bei wachsender Datenflut.

IOP bietet direkte Unterstützung für die **Integration** von Fremdkomponenten, ob Ftp-Server oder Billing-System. Ein weiteres Mini-Framework kapselt gängige Integrationskonzepte und stellt vorgefertigte Aktivitäten bereit wie etwa den Kontaktaufbau zu einem Ftp-Server.

IOP bietet als Framework einen hohen Grad an Wiederverwendbarkeit auf Infrastrukturebene – sei es mit Persistenz, Kommunikation, Verteilung, seinen Design Patterns oder seiner Gesamtarchitektur. Das Prinzip "Beschreiben, Konfigurieren, Generieren" trägt ein Übriges zur Steigerung von **Produktivität** und **Qualität** bei. Die Beschreibung erfolgt über visuelle Modellierung mit UML, die Konfiguration über XML-basierte Werkzeuge und die Generierung mit eigens entwickelten Code-Generatoren für Java und SQL99. Konfiguration wirkt dabei als Mittler und erlaubt es, die visuell modellierte Design-Information auf die jeweils passendsten Technologien und Ansätze abzubilden. Durch die architektur- und use-case-getriebene Vorgehensweise bei der Anwendungsentwicklung sowie die bereitgestellten Richtlinien und Werkzeuge liefert IOP einen klaren Leitfaden für den Projektaufbau. Insbesondere kann dem heutigen Problem der **Expertenknappheit** durch effizientes, rollenbezogenes Coaching von wenig erfahrenen Entwicklern auf wohldefinierten Projektrollen begegnet werden.

Aufbauend auf IOP werden wir einen **Komponentenmarktplatz** etablieren, über den zugeschnittene Komponenten zum Austausch bzw. Kauf und Verkauf angeboten werden können. Auch hier greift der fraktale Gedanke - IOP wird sowohl den Rahmen für die Entwicklung zugeschnittener Komponenten liefern als auch zur Umsetzung des Marktplatzes direkt herangezogen.