

Zukunftsperspektiven

Was kommt nach OPC?

OPC hat sich als Standard zur PC-Anbindung in der Automatisierung etabliert. Ralph Langner, Vorstand der Langner Communications, kommentiert gravierende Veränderungen der IT-Landschaft, die das Weltbild, OPC sei die endgültige Antwort auf alle Fragen der vertikalen Integration, seiner Ansicht nach verrücken.

Wie lange DCOM als technische Basis für OPC noch verfügbar sein wird, ist nach der Einführung von „.NET“ ungewiss. Sicher ist hingegen, dass sich Web Services und andere auf XML basierende Schnitt-



Ralph Langner ist Vorstand der Langner Communications AG in Hamburg

stellen für verteilte Anwendungen immer stärker durchsetzen. Eine Google-Abfrage mit dem Suchbegriff „Web Service“ bringt nahezu doppelt so viele Treffer wie „DCOM“. Nun beschäftigt sich ein Arbeitskreis der OPC Foundation seit immerhin drei Jahren damit, eine Web-Service-Schnittstelle für OPC zu spezifizieren (OPC-XML). Kurios daran ist, dass so eine Schnittstelle OPC in der bisher bekannten Form im Grunde obsolet macht, da der Sinn eines Web Services gerade darin besteht, den „hinter“ dem Interface liegenden Dienst völlig zu kapseln. Anders ausgedrückt: Ob hinter so einem Web Service dann tatsächlich ein OPC-Server läuft oder etwas anderes, ist nicht feststellbar und letztendlich auch egal.

Zu einem Web Service gehört eine Schnittstellendefinition in der so genannten Web Service Definition Language (WSDL), die automatisiert per HTTP geladen und ausgewertet werden kann. Diese Schnittstellendefinition ermöglicht einem Client-Prozess, mit beliebigen Web Services sinnvoll zu arbeiten. Folgerung: Bei Web Services ist die Definition einer einheitlichen Schnittstelle nicht erforderlich, da es bereits eine Metasprache zur Schnittstellenbeschreibung gibt. Eine Visualisierung mit Web-Service-Schnittstelle kann per OPC-XML genau so schwer oder so einfach Prozessvariablen einlesen wie von jedem anderen Web-Service-Interface auch. Somit ist fraglich, wofür ein „Standard“-OPC-XML gebraucht wird.

Was sicher nützlich wäre, aber von der OPC Foundation nicht adressiert wurde, ist eine XML-Schnittstelle, die ohne Web Services – sprich: ohne SOAP – auskommt. SOAP hat eine Reihe von Merkmalen, die es für den industriellen Einsatz nicht gerade empfehlenswert macht. Z. B. ist SOAP, solange es über HTTP läuft, nicht in der Lage, spontane Benachrichtigungen zu transportieren. Des Weiteren ist der Aspekt Zugriffsschutz bei SOAP gelinde gesagt klärungsbedürftig. Wer sich darüber freut, dass SOAP durch jede Firewall geht, muss Netzwerkadministratoren und Firewall-Hersteller für sehr dumm halten. Wir werden bald Firewalls haben, die SOAP komplett filtern – oder zumindest selektive Dienste.

Description	Status	Version	Date
Specifications			
OPC XMLDA 1.00 Specification 0.18	Draft	1.00.0.18	2002-06-13
Software			
OPC XMLDA 1.00 Redistributable	Beta	1.00.1.00	2002-05-10
OPC XMLDA 1.00 Sample Code	Beta	1.00.1.00	2002-05-10
Compliance Tests			
Not available	-	-	-

**Seit einem Jahr unverändert, aber noch nicht veröffentlicht:
Die Spezifikation für OPC-XML**

OPC fehlen entscheidende Middleware-Eigenschaften

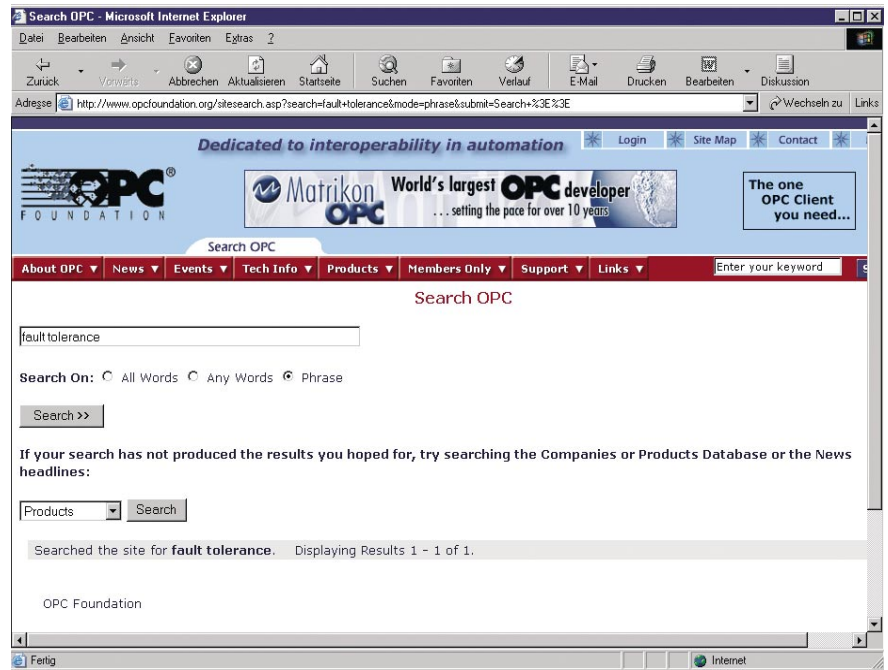
Datenintegration wird künftig nicht mehr auf Zellebene, sondern unternehmensweit betrieben. Das bedeutet, dass immer mehr Software-Anwendungen an den Daten der Automatisierungslandschaft interessiert und auch von diesen Daten abhängig sind. Hieraus ergeben sich im Hinblick auf OPC mehrere Probleme:

- OPC ist de facto auf Windows-Systeme beschränkt und stellt dadurch eine Barriere dar für Anwendungen, die auf einem anderen Betriebssystem laufen, wie Leitsysteme auf Basis von Unix und VMS,
- der Zugriffsschutz bei OPC ist unzureichend, da allein auf DCOM basierend,
- OPC beinhaltet keinerlei Unterstützung für einen fehlertoleranten Betrieb und
- OPC basiert auf einer nicht kaskadierbaren 2-Tier-Architektur (es ist nicht vorgesehen, dass ein OPC-Server Daten von anderen OPC-Servern sammelt und aufbereitet, sondern alle Prozesslogik muss in der Anwendung implementiert werden).

Am gravierendsten für das Szenario der Zukunft ist möglicherweise Punkt Drei: Fällt ein OPC-Server aus (sei es aufgrund von Programmfehlern, Netzwerkproblemen, Stromausfall oder Ähnlichem), erhält keiner der angemeldeten Clients noch Daten. Solange diese Daten nur von einer Visualisierung benötigt werden, hält sich der Schaden in Grenzen. Sind mehrere nachgelagerte Anwendungen von der Lagerlogistik bis zum Vertrieb plötzlich „blind“, kann das teuer werden. Mit zunehmendem Vernetzungsgrad wird eine hochverfügbare Architektur ein Muss.

Das fehlende Bindeglied auf der „letzten Meile“

Bei seiner Einführung kurierte OPC die Symptome eines Problems, nicht aber für deren eigentliche Ursache. OPC macht Schluss mit unterschiedlichen Treiberarchitekturen, nicht jedoch mit dem Nebeneinander unterschiedlicher Industrieprotokolle. Die „letzte Meile“ zur Automatisierungswelt ist trotz oder gerade wegen OPC nach wie vor sehr heterogen. Dank OPC steht es jedem Hersteller von Automatisierungskomponenten prinzipiell frei, sein eigenes Industrieprotokoll zu „stricken“. So lange ein OPC-Server dazu geliefert wurde, war für den Hersteller der Visualisierung die Welt in Ordnung.



Nach dem Stichwort „Fehlertoleranz“ sucht man bei OPC vergeblich

Nun war die Vielfalt von Protokollen und Medien vor wenigen Jahren noch ein Übel, das man eigentlich nur als naturgegebenes Schicksal hinnehmen konnte. Nachdem sich Ethernet und TCP/IP als dominantes Medium für den PC-Datenaustausch durchgesetzt hat, ergibt sich ein neues Bild. Erstens ist es nicht nur technisch möglich, sondern auch hochgradig sinnvoll, ein einheitliches Anwendungsprotokoll auf Basis von TCP/IP zu verwenden. Zweitens ist ein solches modernes Protokoll aus Performance-Gründen zwingend erforderlich. Alles, was an

Modbus, basiert auf Polling-Verfahren und ist allein deshalb bereits für TCP/IP technisch ungeeignet. Drittens ist die Funktionsarmut gängiger Industrieprotokolle grotesk, denken wir z. B. an „Brot-und-Butter“-Funktionalitäten aus dem Consumer-Bereich wie Plug-and-Play-Konfiguration.

Es ist also dringend an der Zeit, Herstellern als auch Anwendern ein leistungsfähiges und geräteunabhängiges Protokoll für den PC-Datenaustausch zu geben. Wird ein solches Protokoll PC-seitig über einen OPC-Server betrieben, ändert sich für das Anwendungsprogramm an der Software-Schnittstelle nichts, jedoch wird der Datenaustausch wesentlich performanter, und die Datenstruktur des OPC-Servers könnte automatisch aus den Projektierungsinformationen erstellt werden. Allerdings: Setzt sich ein solches Schicht-7-Protokoll auf breiter Front durch, dann würde damit auch der Hauptgrund für den Einsatz von OPC entfallen, nämlich die Medien- und Protokollvielfalt.

Blaupause einer Systemarchitektur der Zukunft

Die wichtigsten Eckpunkte einer Systemarchitektur für die vertikale Integration der Zukunft sind:

Verwendung von Web-Interfaces. XML und Web Services sind die Schnittstellen für verteilte Anwendungen im IT-Bereich. Sie müssen auch für den Zugriff auf Prozessvariablen genutzt werden können. Es lässt sich auch nicht länger ignorieren, dass weite Bereiche moderner Unternehmens-

Abkürzungsdefinitionen

DCOM	– Distributed Component Object Model
EAI	– Enterprise Application Integration
HTTP	– Hypertext Transfer Protocol
IT	– Informationstechnologie
OPC	– OLE for Process Control
SOAP	– Simple Object Access Protocol
TCP/IP	– Transmission Control Protocol/Internet Protocol
XML	– Extensible Markup Language

althergebrachten Protokollen von den früher dominanten Punkt-zu-Punkt-Schnittstellen übernommen wurde, wie

Software nicht in Visual Basic, sondern in Java geschrieben werden.

Plattformunabhängigkeit. Die Beschränkung einer Treiberarchitektur auf MS-Windows ist dort kein Problem, wo es vornehmlich um die „Killer-Anwendung“ HMI geht (praktisch alle Visualisierungen laufen unter Windows). Liegt das Augenmerk jedoch auf einer unternehmensweiten Vernetzung, müssen weitere Betriebssysteme unterstützt werden – allen voran Linux.

Kaskadierbares Datenmodell. Wenn ein Gerätetreiber seine Daten immer nur an eine High-Level-Anwendung liefern kann (2-Tier-Modell), dann bedingt dies, dass alle Prozesslogik in die Anwendung implementiert werden muss. Früher oder später werden mehrere Anwendungen die gleichen Algorithmen implementieren. Für die vertikale Integration ist nur ein N-Tier-Modell angemessen, in dem z. B. Kennzahlen und Statistiken nicht von Anwendungen, sondern von Software-Agenten berechnet werden.

Fehlertoleranz. Der Zugriff auf Betriebs- und Maschinendaten muss fehlertolerant ausgelegt werden können, so dass alltägliche Netzwerkprobleme, Rechnerabstürze usw. nicht zum Datenverlust führen. Dies bedingt redundant ausgelegte Systeme, die sich gegenseitig überwachen.

Zugriffskontrolle auf Benutzer- und Gruppenebene. Für den Zugriff auf Prozessvariablen müssen in einfacher Weise Rechte vergeben werden können. Dies muss auf der Basis von authentifizierten Benutzern und deren Zugehörigkeit zu Gruppen (z. B. Produktion, Wartung, Vertrieb, ...) erfolgen können.

Plug-and-Play. Es ist absurd, dass Plug-and-Play-Techniken, die im Bereich der Consumer-Elektronik inzwischen selbstverständlich sind, noch keinerlei Zugang zur Industrieautomation gefunden haben. Hier wird nach wie vor doppelt und dreifach konfiguriert, was nicht nur teuer, sondern auch fehleranfällig ist. Technische Hinderungsgründe hierfür gibt es nicht.

Leistungsfähiges Protokoll zum Feldgerät. TCP/IP hat andere Charakteristika als Punkt-zu-

Punkt-Medien und Feldbusse. Die Integration eines modernen, schlanken Protokolls zur Automatisierungsperipherie hin, welches ereignisgesteuert arbeitet, ist unverzichtbar, wenn es mittelfristig nicht zum Netzwerk-Gau kommen soll.

Um sich ein Bild von einer möglichen Systemumgebung der Zukunft zu machen, lohnt sich ein Blick auf moderne IT-Integrationsplattformen. Unter den

Stichworten EAI finden sich dort Middleware-Architekturen, die im Kern ein ähnliches Problem lösen: unternehmensweit zuverlässig eine Vielzahl unterschiedlicher Systeme miteinander zu verbinden. Die Connectivity zu einem Mainframe-Dinosaurier per SNA und 3270 ist im Grunde der Anbindung einer Simatic S5 ähnlich, und von ihr kann man lernen.

Ralph Langner