
Einbindung von Simulationswerkzeugen in Datenerfassungs-, Datenvisualisierungs- und Steuerungssoftware

Karin Glader^a, Simon Kranzer^a, Sebastian Schöndorfer^b, Roland Graf^a,
Karl Entacher^a, Robert Merz^a, Reinhard Mayr^b

^a FH Salzburg GmbH, Urstein Süd 1, A-5412 Puch/Hallein, AUSTRIA

^b Ing. Punzenberger COPA-DATA GmbH, Karolingerstraße 7b, A-5020 Salzburg, AUSTRIA

KURZFASSUNG/ABSTRACT:

Der Beitrag beschäftigt sich mit der Integration von Software-Werkzeugen zur Simulation von Industrieanlagen bzw. industriellen Prozessen, mit einem Hauptfokus auf diskrete eventbasierte Simulation, in ebenfalls Software-basierten Systemen aus dem Bereich Supervisory Control and Data Acquisition (SCADA) sowie Human-Machine Interface beziehungsweise Human-Machine Interaction (HMI). Anhand einer Beispielimplementierung basierend auf der HMI/SCADA-Lösung zenon der Firma COPA-DATA und auf der Mehr-Methoden-Simulationssoftware AnyLogic der The AnyLogic Company wird demonstriert, wie und mit welchen Methoden und Datenmodellen eine Simulation in das Überwachen und Steuern technischer Prozesse mittels eines Computer-Systems integriert werden kann. Der dabei entwickelte Software-Adapter ist eine Zusammenfassung von bewährten Technologien zu einer neuartigen einheitlichen Programmierschnittstelle (API) auf Basis der Simple Object Access Protocol (SOAP). In einem Wechselspiel zwischen Hardware-In-The-Loop und Software-In-The-Loop Szenario können durch die Integration von Simulation und Echtssystemen Zustände von Sensoren und Aktoren der Anlage mit den Zuständen von Sensoren und Aktoren der Simulation abgeglichen werden, um einerseits das Modell ständig zu erweitern bzw. zu verbessern, und andererseits Aussagen über Stabilität und Verhalten des Realsystems zu erlauben. Zusätzlich werden im Rahmen des Beitrages bereits umgesetzte und auch zukünftige Anwendungen einer derartigen Verbindung erläutert und diskutiert.

1 EINLEITUNG

Unter dem Sammelbegriff „Software für digitale Produktion“ existieren Programme und Rahmenwerke zur Vereinfachung, Reproduktion und Automatisierung von Planung, Inbetriebnahme, Wartung, Analyse und Visualisierung von allen Teilen der Produktionsprozesskette. Insbesondere große Unternehmen und Betriebe in hochautomatisierten Industrien, wie etwa der Automobil- oder Energiebranche, setzten derartige Werkzeuge schon sehr erfolgreich ein. Trotzdem existieren noch eine Reihe von Branchen, die weniger Automatisierungswerkzeuge nutzen, wie etwa die Sägeindustrie sowie viele Klein- und Mittelunternehmen, die aus den verschiedensten Gründen (Kosten, Ausbildungsstand des Personals, fehlende Unterstützung bei der Auswahl der geeigneten Werkzeuge) keine oder nur teilweise computergestützte Instrumente einsetzen. Gerade Simulationswerkzeuge werden dabei besonders selten erfolgreich eingesetzt. Im Gegensatz dazu verfügen oft auch weniger IKT-affine Klein- und Mittelunternehmen bereits über Software zur Anlagensteuerung und zur Aufzeichnung und Visualisierung von Messwerten.

Eine Kopplung zwischen SCADA-Systemen und einer Simulationsumgebung wird auch im Kontext der System-Sicherheit betrachtet. In [1] und [2] wird die Implementierung einer Simulationsumgebung zur Analyse der Sicherheit in SCADA-Systemen beschrieben mit dem Ziel, Methoden zur Analyse und Bewertung der Auswirkungen von Angriffen auf die Sicherheit eine SCADA-Systems zu entwickeln [1]. Der Fokus der aktuellen, publizierten Forschungen liegt in der Simulation von SCADA-Systemen. Im Bereich der Einbindung von Software, die Prozesse

einer realen Anlage simuliert, sind zum Zeitpunkt der Durchführung keine Erfahrungswerte bekannt.

2 INTEGRATION VON SIMULATIONSWERKZEUGEN IN SCADA/HMI-SYSTEMEN

Um eine Lösung für die in Abschnitt 1 definierte Problemstellung zu erarbeiten, ist die Integration von Simulationen und –werkzeugen in Systemen zur Planung, Steuerung und Visualisierung von Produktionsanlagen (SCADA) und die Entwicklung von Strategien zum Einsatz dieser zur Wartung, Fehleranalyse und Optimierung, Ziel des vorliegenden Beitrages.

Zur Validierung der Vorgehensweise zur Einbindung einer Simulationsumgebung in ein SCADA System wurden zenon [3] und AnyLogic [4] ausgewählt. Die verwendete Version AnyLogic 7 ist eine plattformübergreifende Software, die eine grafische Modellierungssprache beinhaltet und die eine Möglichkeit bietet, Simulationsmodelle mit Java-Code zu erweitern. Im Gegensatz zu anderen Simulationsumgebungen werden hierbei keine genauen Programmierkenntnisse vom Anwender gefordert, da sich die meisten grundlegenden Funktionen parametrisieren anstatt programmieren lassen. Durch AnyLogic werden Prozesse einer realen Anlage durch Simulationen ersetzt. Der Steuerungsalgorithmus soll unverändert bleiben, da ansonsten die Anbindung an ein SCADA-System keinen Vorteil gegenüber einer reinen Simulation bieten würde. Prozesse in Zeitraffer ablaufen zu lassen wurde dadurch ausgegrenzt. Während der Konzepterstellung wurden die verschiedenen Anbindungsmöglichkeiten von AnyLogic an zenon analysiert. Die Möglichkeit, AnyLogic über die zenon eigene Soft-SPS einzubinden wurde auf Grund der in ersten Tests erreichten geringen Kommunikationsgeschwindigkeiten außer Acht gelassen. Die Lösung ist zudem nur auf zenon angepasst und kann für andere SCADA Systeme nicht verwendet werden. Die Programmierung eines eigenen Treibers wurde wegen des erheblichen Entwicklungsaufwands bezüglich Programmierumgebung, Testing, etc., sowie wegen der Defizite im Bereich der Wiederverwendbarkeit ausgeschlossen. Durch die in zenon integrierte Schnittstelle der Visual Studio Tools for Applications (VSTA) [5] und eines eigens dafür entwickelten Server, über den ein Datenaustausch stattfindet, soll eine Anbindung einer Simulationssoftware an zenon ermöglicht werden. Dadurch ist die Einbindung nicht speziell an zenon angepasst und für zukünftige Szenarien wiederverwendbar.

Zur Erreichung der gewünschten Ergebnisse wurde das in Abbildung 1 dargestellte Systemmodell entwickelt. Anhand dieser Best-Practice-Vorgehensweise können zukünftig auch andere als die beispielhaft implementierten SCADA/HMI und Simulationen integriert werden.

3 TECHNISCHE UMSETZUNG

Die Integration, Abbildung 1, erfolgt über die integrierte VSTA-Schnittstelle. Durch die Verwendung der VSTA Schnittstelle und dem darunterliegenden .NET Framework [6] wurde ein WCF-Webservice [7] als Kommunikationsframework gewählt.

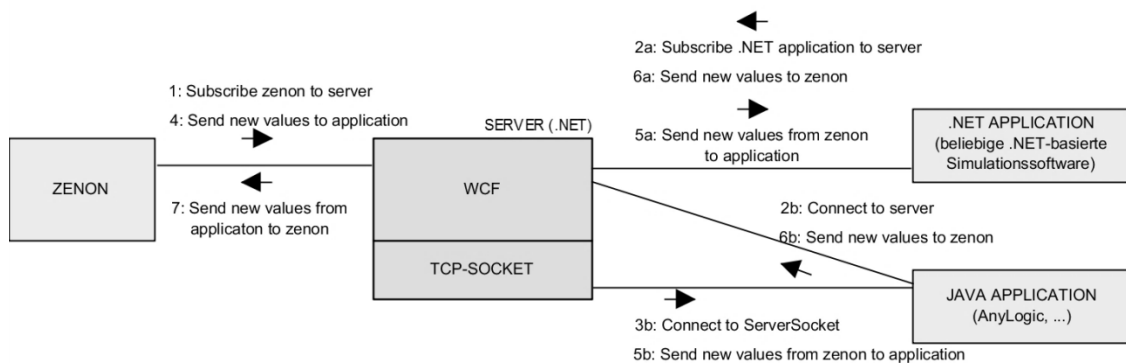


Abbildung 1. Systemabbildung der Beispielintegration von HMI/SCADA und Simulation

Die Realisierung beruht somit auf einer Zusammenfassung von bewährten Technologien zu einer einheitlichen Programmierschnittstelle auf SOAP-Basis. Ein WCF-Server bietet zudem den Vorteil, dass mehrere Transportprotokolle zur Datenübertragung wie http, tcp oder MSMQ unterstützt werden [8].

Der Versuchsaufbau soll zeigen, dass der Datenaustausch zwischen zenon und einer Simulationssoftware möglich ist, da dies die Grundlage ist, um Zustände von Sensoren und Aktoren der Anlage mit den Zuständen von Sensoren und Aktoren der Simulation abzugleichen. Als Testsoftware dient die auf Java basierende Prozesssimulationssoftware AnyLogic. AnyLogic ist ein Simulationswerkzeug, das diskret-ereignisorientierte, agentenbasierte und systemdynamische Modellierung unterstützt. Beim Verbindungsaufbau zwischen einer auf Java basierten Simulationssoftware und zenon über einen WCF-Server wird die Java API for XML Web Services (JAX-WS) benötigt. Die Verwendung eines WCF-Servers als Schnittstelle ermöglicht es, unterschiedliche Plattformen miteinander zu verbinden.

Wie in Abbildung 1 dargestellt, werden für die Einbindung einer Simulationssoftware in eine SCADA/HMI drei Hauptkomponenten benötigt. Die SCADA-Software, die Simulationssoftware, sowie ein Server, der als Schnittstelle zwischen Realität und Simulation fungiert.

Der entwickelte Server basiert auf der Windows Communication Foundation (WCF). WCF beschreibt eine Kommunikationsplattform zur Entwicklung von dienstbasierten Anwendungen in .NET und fasst vorgehende Technologien zu einer einheitlichen Programmierschnittstelle auf SOAP-Basis zusammen. Die Übertragung von Daten verläuft bei WCF-Anwendungen über einen oder mehrere Endpunkte des Dienstes. Diese ermöglichen den Zugriff auf die vom Server angebotenen Funktionalitäten durch externe Anwendungen. Bei der Konfiguration von Endpunkten wird das sogenannte ABC (Address, Binding, Contract) Prinzip angewandt. Da definierte Funktionalitäten mit einem Endpunkt zusammenhängen, wird jedem Endpunkt eines Webservices eine eindeutige Adresse (URI) zugeteilt, über welche die zur Verfügung stehenden Funktionen von anderen Anwendungen abgerufen werden können. Das Binding beinhaltet Informationen zum verwendeten Transportprotokoll und zur Codierungsart. Jedem Binding wird wiederum ein Vertrag zugeteilt, der die Funktionalität und falls benötigt auch benutzerdefinierte Datentypen eines Dienstes definiert [6].

3.1 Kommunikation zwischen Realität und Simulation

Die Grundlage, um Daten und Informationen zwischen einer Anlage und deren Simulation auszutauschen, ist die bidirektionale Kommunikation. Beiden Seiten muss es möglich sein, Daten ohne explizite Nachfrage an die jeweilige Software weiterleiten zu können. Durch ständiges Pollen wird eine hohe Netzwerkbelastung ausgelöst, identische Daten würden mehrfach verarbeitet werden, was zu Performanceeinbrüchen führen kann.

Eine beidseitige Kommunikation zwischen dem WCF-Server und einer .NET basierten Anwendung wird über das vordefinierte WSHttpDualBinding ermöglicht. Der Server stellt Methoden zur Verfügung, die eine Anwendung nach Einbinden des Webservices zur Datenübermittlung aufrufen kann (siehe Abbildung 1, Pfad 4). Zudem werden die im Server definierten Callback-Methoden festgelegt. Diese Callback-Methoden können jederzeit vom Server aufgerufen werden, um neue und aktuelle Daten zurück an die .NET Anwendung zu senden.

Die Verwendung eines WCF-Servers über Java basierte Anwendungen ist in der Regel deutlich aufwendiger. Das einzige von Microsoft vordefinierte Binding, das über Java Anwendungen verwendbar ist, ist das BasicHttpBinding ([7], [9]), das laut Definition aber keine beidseitige Kommunikation vorsieht. Um eine zumindest einseitige Kommunikation von Java basierten Anwendungen zum WCF-Server zu ermöglichen, werden über die Java API for XML Web Services (JAX-WS) Kommunikationsstubs erzeugt. Über den angeführten Aufruf des JAX-WS Tools wsimport in der Kommandozeile werden Informationen anhand der bestehenden WSDL-Daten des WCF-Servers in definierte Dateien gespeichert.

Die erzeugten Dateien beinhalten die Informationen, die zur Kommunikation zum Server benötigt werden, wie beispielsweise Name, Namespace, WSDL-Adresse oder die am WCF-Server verfügbaren Methoden. Mit Hilfe dieser Dateien ist es möglich, diese Methoden innerhalb der Java Anwendung aufzurufen. Für die Kommunikationsrichtung vom Server zur Java Anwendung wird zusätzlich ein TCP-Socket verwendet [10]. Beim Starten der Anwendung wird der Socket geöffnet. Sobald sich die Anwendung das erste Mal mit dem Server verbindet, initiiert der Server den Verbindungsaufbau des Rückkanals. Über einen eigenen Thread wird im TCP-Socket ständig auf Daten des Servers gewartet, welche im Anschluss konvertiert und für die Anwendung bereitgestellt werden. Sobald sich die Werte der Simulation verändern, werden diese Informationen über das BasicHttpBinding an den Server übermittelt, der anschließend die Zustandsänderungen über das WSDualHttpBinding an die SCADA-Software weitersendet.

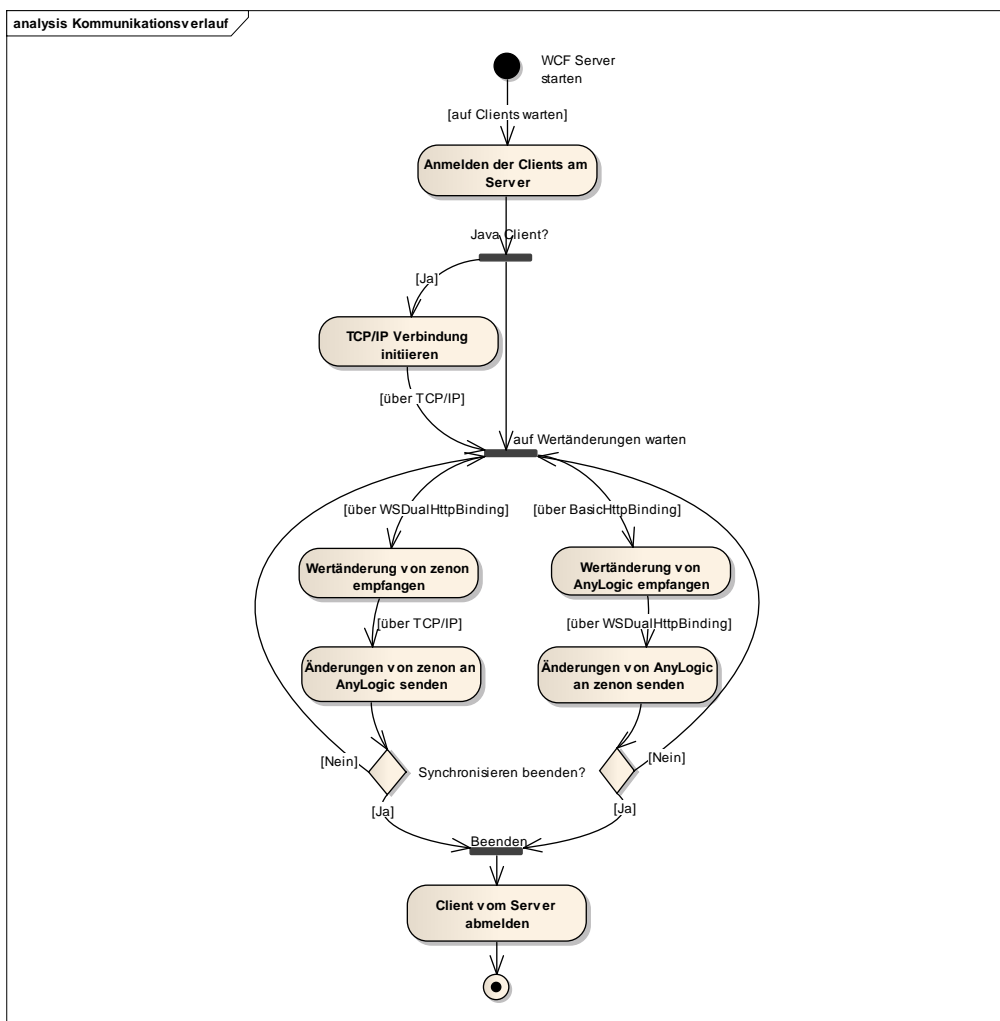


Abbildung 2. Kommunikation über den WCF-Server und einen TCP-Socket

Die Abbildung 2 verdeutlicht den Kommunikationsablauf bei Datenänderungen zwischen zenon und einer Java basierten Simulationsumgebung wie AnyLogic und zeigt den internen Ablauf des WCF- Servers als zentralen Vermittlungspunkt.

3.2 Ergebnisse

Simulationen bieten den Vorteil, die Kosten eines Unternehmens zu senken, da Veränderungen frühzeitig überprüfbar sind und dadurch auch das generelle Risiko für Neuanlagen gesenkt

werden kann. Fehlerzustände können über eine Simulation nachgestellt und somit frühzeitig detektiert werden. Wenn bereits bekannt ist, wie sich Fehlerzustände ereignen bzw. wie sich diese in Anlagenzustände ausdrücken, können Abweichungen bereits vor Eintreten des Problems ermittelt werden.

Die sprach- und plattformunabhängige Integration von Simulationswerkzeugen in HMI/SCADA-Systeme wie zenon kann durch die Implementierung einer XML-basierter Kommunikation und Verwendung von Rahmenwerken wie etwa der WCF realisiert werden. Es entsteht eine flexible Anbindung zu Simulationssoftware. Zusätzlich konnte durch die bei der Beispielimplementierung gewählten Technologien gezeigt werden, dass für Simulationsumgebungen die auf .NET basieren durch wenig Mehraufwand beispielsweise eine Lastenaufteilung realisiert werden kann und verschiedene Teile einer Anlage auf verteilten Systemen als in sich geschlossenes System laufen können.

Aufbauend auf die Versuche, aktuelle Daten bzw. Zustände einer Anlage von zenon über WCF an Simulationsumgebungen zu senden, wurde ein weiteres Anwendungsszenario entwickelt. Dabei werden die Daten an eine C# Applikation gesendet, welche über eine bestimmte Zeit den Ablauf der Industrieanlage beobachtet und somit die Zustandsänderungen am System mitlernt. Denkbar ist eine Anwendung, welche als selbstlernender Simulator agiert und damit nicht nur Simulationsaufgaben übernimmt, sondern auch Abweichungen zwischen realem Prozess und Simulation erkennen kann. Dadurch stellt das Programm eine Art Kontrollinstanz zum eigentlichen Prozess dar. Fehler und Abweichungen zum korrekten Ablauf können somit frühzeitig erkannt werden und Wartungsarbeiten eingeplant werden.

In einem Folgeprojekt, das sich mit der Identifikation, Planung und Unterstützung von Wartungsarbeiten beschäftigt, wird die SCADA/HMI-Simulationskopplung dazu benutzt, Abweichungen einer laufenden Anlage vom Sollzustand automatisch zu erkennen und entsprechende Hinweise oder Gegenmaßnahmen anzustoßen.

LITERATURVERWEISE

- [1] Wang, Chunlei, Fang, Lan, Dai, Yiqi (2010): A Simulation Environment for SCADA Security Analysis and Assessment. In: 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Vol. I, S. 342 - 347
- [2] Querioz, Carlos, Mahmood, Abdun, Tari, Zahir (2011): SCADASim – A Framework for Building SCADA Simulations. In: IEEE Transactions on Smart Grid, Vol. 2, No. 4, S. 589-597
- [3] Ing. Punzenberger COPA-DATA GmbH: COPA-DATA, [Online]. Available: <http://www.copadata.com>. [Zugriff am 14.10.2014].
- [4] The AnyLogic Company: AnyLogic, [Online]. Available: <http://www.anylogic.de/>. [Zugriff am 14.10.2014].
- [5] Microsoft Corporation: „Visual Studio Tools for Application,“ [Online]. Available: <http://blogs.msdn.com/b/vsta/>. [Zugriff am 14.10.2014].
- [6] Microsoft Corporation: Microsoft .NET, [Online]. Available: <http://www.microsoft.com/net>. [Zugriff am 14.10.2014].
- [7] Microsoft Corporation: What Is Windows Communication Foundation, [Online]. Available: <http://www.microsoft.com/net>. [Zugriff am 14.10.2014].
- [8] Tharun, Kola, Mupalla, Prudhvi, Reddy, Satti (2013): Advantages of WCF over Web Services. In International Journal of Computer Science and Mobile Computing, Vol. 2, S. 340 - 345
- [9] Peiris, Chris., Mulder, Dennis (2007): Pro WCF: Practical Microsoft SOA Implementation. Springer Verlag, New York
- [10] Calvert, Kenneth, Donahoo, Michael (2008): TCP/IP Sockets in Java: Practical Guide for Programmers. Elsevier, Burlington