

Beschreibung automatisierungstechnischer Simulationsobjekte mit XML

Dr.-Ing. Hartmut Weber
SimVA Software & Services GmbH, Gronau

1. Hintergrund

Auch im Sondermaschinenbau wird zunehmend versucht, bereits entwickelte Komponenten für ähnliche Aufträge wieder zu verwenden, um so den Engineering-Aufwand möglichst gering zu halten [2]. In diesem Zusammenhang ist unter einer Komponente eine mechatronische Einheit zu verstehen, bestehend aus mechanischen, elektrischen und steuerungstechnischen Teilen.

Neben der eigentlichen mechatronischen Funktionalität, die durch Kräfte, Momente und Bewegungen beschrieben werden kann, spielt insbesondere auch die Bedienbarkeit der mechatronischen Komponente eine wichtige Rolle. Diese wird geprägt durch die Bedienelemente und die Aktionen und Reaktionen des Bedieners, um bestimmte Betriebszustände der Maschine zu erreichen. Als Bedienelemente kommen je nach Einsatzziel einfache und robuste Taster und Leuchtmelder oder auch deutlich komplexere Strukturen, wie etwa Bedien-PC mit Touchscreen-Oberflächen, in Betracht.

Die Wirkung der Bedienelemente auf das Maschinenverhalten wird häufig durch SPS-Programme bewerkstelligt, die Bedienzustände mit Maschinenzuständen verknüpfen. Zur Wiederverwendung einer mechatronischen Komponente gehört also neben der mechanischen und elektrotechnischen Integration in die neue Maschine auch die steuerungstechnische Integration, so dass die Anforderungen hinsichtlich der Bedienbarkeit der neuen Maschine erfüllt werden.

Um die Integrationsleistungen zu einem möglichst frühen Zeitpunkt im Engineering-Prozess überprüfen zu können, ist es sinnvoll, Simulationsmethoden anzuwenden [4]. Im Bereich des Maschinenbaus führt der Einsatz einer steuerungstechnischen Simulation zum Nachweis der Qualität einer Steuerungsentwicklung bereits vor der ersten Inbetriebnahme der Maschine und damit zu einer Verringerung des eigentlichen Inbetriebnahmeaufwands.

Da in den Engineering-Prozessen des Maschinenbaus bereits eine Reihe von Engineering-Werkzeugen standardmäßig eingesetzt wird, spielt der Aspekt der Integration in die Engineering-Prozesskette für ein Simulationswerkzeug eine entscheidende Rolle. Anwender profitieren nur dann von einem Simulationswerkzeug, wenn sein Nutzen ohne merklichen Aufwand für eine zusätzliche Anwendungs- oder Prozessintegration abgerufen werden kann.

Einen Ansatz für eine möglichst einfache Prozessintegration bietet XML als Metasprache zur Spezifikation von Dokumentengrammatiken (Bild 1). Durch den einfachen lexikalischen Aufbau (ASCII) und die Vielzahl verfügbarer Tools bietet der Einsatz von XML als quasi universelles Datenformat viele Vorteile.

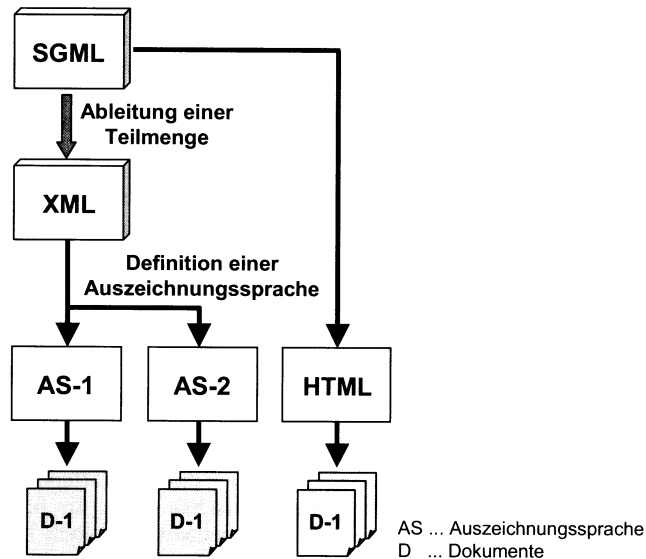


Bild 1: Spezifikation von Dokumentengrammatiken mit XML (aus [3])

2. Anforderungen an das Simulationswerkzeug

Ein Werkzeug zur Prüfung der Wiederverwendungsergebnisse mechatronischer Komponenten nebst der dazugehörigen Steuerungstechnik muss den folgenden Anforderungen genügen:

a) Integration in bestehende Prozessketten

Bestehende Prozessketten dürfen durch den Einsatz des Simulationswerkzeugs nicht gestört werden. Eine Störung könnte etwa darin bestehen, dass für einen Prozessschritt

- ein anderes oder modifiziertes Werkzeug eingesetzt werden muss,
- die Abfolge der bestehenden Prozessschritte modifiziert werden muss oder
- die Schnittstellen zwischen bestehenden Prozessschritten geändert werden müssen.

Diese und ähnliche Störungen sollen nicht geschehen, was positiv unter dem Schlagwort „Offenheit des Simulationssystems“ zusammengefasst werden kann.

b) Vermeidung wiederholter Eingaben

Die für den Simulationslauf notwendigen Daten sollen sich weitestgehend automatisch aus den bisher bereits innerhalb der Prozesskette eingegebenen Daten ergeben. Die Wiederholung der Eingabe von Daten, die bereits zuvor innerhalb der Prozesskette aufgenommen worden sind, ist nicht akzeptabel.

c) Echtzeitfähigkeit

Unter der Forderung nach Echtzeitfähigkeit soll im Zusammenhang mit der Steuerungssimulation verstanden werden, dass die sichtbaren Reaktionen der simulierten Maschine das gleiche Zeitverhalten aufweisen wie die sichtbaren Reaktionen der realen Maschine. Eine solche sichtbare Reaktion kann etwa das sich in Bewegung Setzen einer Achse als Reaktion auf eine bestimmte Bedienaktion sein.

d) Vollständige Protokollierung

Sollen Simulationsläufe als Nachweis für die Qualität der Steuerungssoftware dienen, so müssen alle Benutzereingaben und SPS-Eingangs- und Ausgangssignale in ihrem zeitlichen Ablauf protokolliert werden. Sollte sich zu einem späteren Zeitpunkt ein unerwünschtes Maschinenverhalten zeigen, so kann gegebenenfalls anhand eines Protokolls nachgewiesen werden, dass das unerwünschte Verhalten auf eine unbefugte nachträgliche Änderung des SPS-Programms zurückzuführen ist.

e) Domänenspezifisches Handling

Anwender müssen ihre übliche Arbeitsweise und ihre Erfahrungen im Simulationswerkzeug wieder finden, um dieses effektiv nutzen zu können. Daher ist ein domänenspezifisches Handling wesentlich für den Erfolg eines Simulationswerkzeugs. Konkret bedeutet dies beispielsweise, dass das Werkzeug eine gleiche Sichtweise auf die Realität ermöglichen muss wie die anderen bereits in der Prozesskette eingesetzten Werkzeuge. Dies kann etwa eine objektorientierte Sichtweise sein – falls jedoch andere Werkzeuge eine solche Sicht nicht unterstützen, wäre diese eher hinderlich für die Akzeptanz des Simulationswerkzeugs.

Die Beschreibung automatisierungstechnischer Simulationsobjekte mit XML ist im Hinblick auf die genannten Anforderungen insbesondere für die Integration des Simulationswerkzeugs in bestehende Prozessketten [1] und die hiermit verbundene Vermeidung wiederholter Eingaben von Bedeutung. Die übrigen Anforderungen sind nicht weniger relevant, jedoch primär unabhängig von bestimmten Prozess-Schnittstellen.

3. Prozessintegration durch Nutzung von XML-Schnittstellen

Zur Darstellung der Nutzung von XML für die Integration von Engineering Prozessen wird im Folgenden zunächst beispielhaft das konkrete Werkzeug-Szenario eines Sondermaschinenbau-Unternehmens vorgestellt. Mit Hilfe dieser Werkzeuge werden in dem betrachteten Unternehmen sowohl die Neuentwicklung wie auch die Wiederverwendung bereits entwickelter Maschinenkomponenten realisiert.

Für die mechanische bzw. die elektrotechnische Konstruktion kommen *SolidWorks* bzw. *EPLAN* zum Einsatz (Bild 2). Die Steuerungsentwicklung erfolgt für eine Reihe von Endkunden des Sondermaschinenbauers mit *Step7* und zur Gestaltung der Bedienoberflächen sowie des Bedienpults werden *WinCC* und *Visio* eingesetzt.

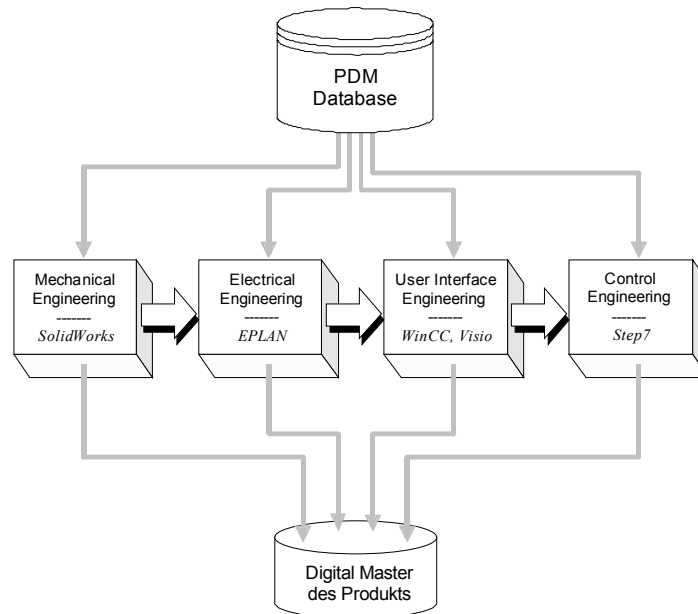


Bild 2: Konkrete Engineering-Prozesskette

Nicht alle genannten Werkzeuge verfügen bereits über XML-Schnittstellen, so dass die Modellgenerierung nur teilweise in der in Bild 3 dargestellten idealtypischen Form erfolgen kann. Beschränkt man sich auf den Aspekt der steuerungstechnischen Simulation, so sind

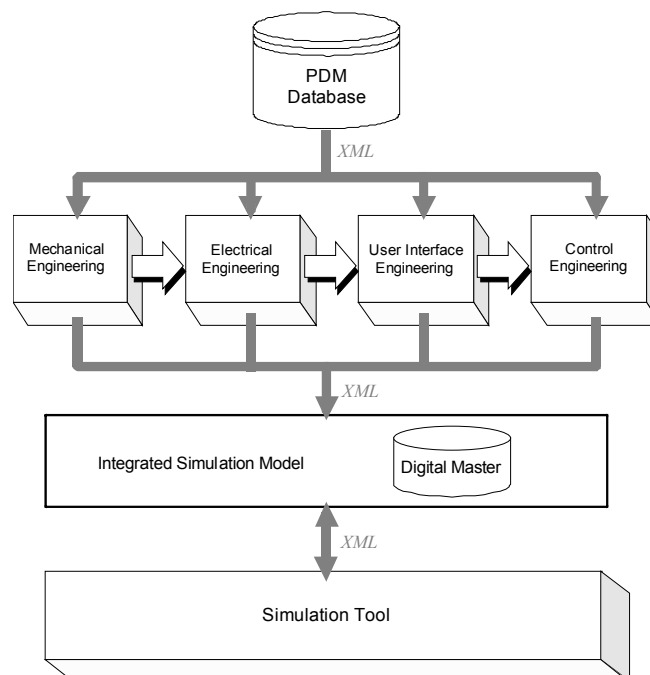


Bild 3: Idealtypische Engineering-Prozesskette mit integrierter Simulation

insbesondere die Werkzeuge zur SPS-Programmerstellung und zur Bedienoberflächen-Gestaltung zu betrachten. Innerhalb der konkreten Prozesskette verfügt die Bedienoberflächen-Gestaltung über eine XML-Schnittstelle, die SPS-Programmerstellung nicht. Daher muss für ein Simulationswerkzeug ein Konzept verfolgt werden, dass die SPS-Simulation als proprietäre Lösung einbettet.

Das von der SimVA GmbH entwickelte IMS-System (Bild 4) benutzt die Werkzeuge *Step7* und *Visio* der bestehenden Prozesskette nach Bild 2. Das Simulationsmodell beinhaltet die aus *Visio* übernommenen Bedienelemente im XML-Format, weitere Simulationsobjekte, die mit Hilfe des Simulationswerkzeugs erstellt wurden, sowie Verbindungen zwischen den Simulationsobjekten. Weiterhin gehört zum Simulationsmodell der SPS-Code, der innerhalb des Simulationswerkzeugs durch einen integrierten SPS-Simulator aus der gegebenen Prozesskette bearbeitet wird.

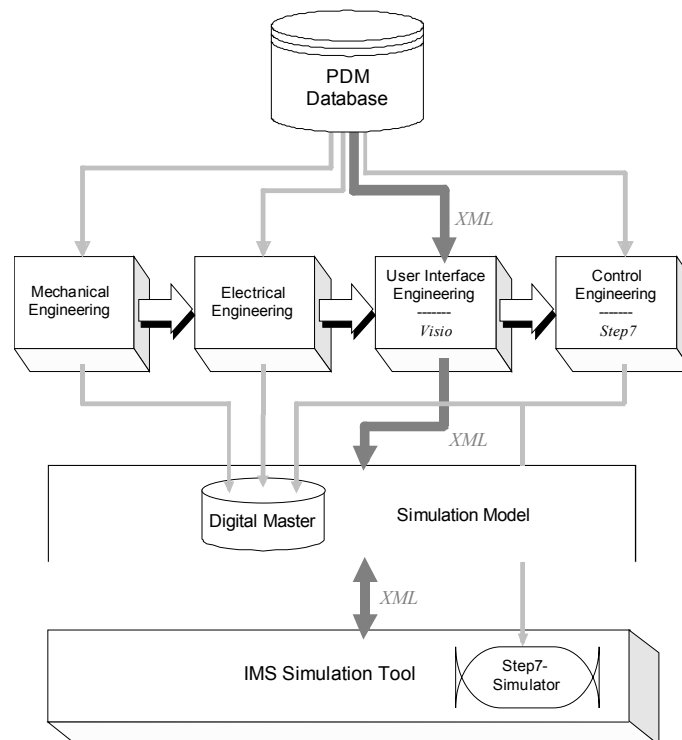


Bild 4: IMS-System

4. Kategorien von Simulationsobjekten

Abhängig von seiner Komplexität kann jedes der IMS-Simulationsobjekte einer von drei möglichen Kategorien zugeordnet werden:

a) Einfache Simulationsobjekte

Zu den einfachen Simulationsobjekten gehören die Bedienelemente (Bild 5) und die Logik-elemente (Bild 6). Letztere werden benötigt, um Komponenten darzustellen, die nicht direkt aus der Prozesskette übernommen werden können. Dies gilt etwa für das Electrical En-

gineering: Mangels verfügbarer Schnittstellen muss mit Hilfe der Logikelemente das Verhalten des Schaltschranks separat modelliert werden.

b) Zusammengesetzte Simulationsobjekte

Aus mehreren Logikelementen entstehen durch Verschaltung zusammengesetzte Simulationsobjekte. Bild 7 zeigt beispielsweise eine „speichernde Einschaltverzögerung“, die folgen-

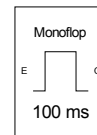
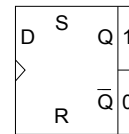
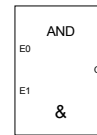
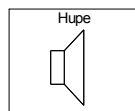
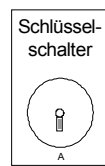
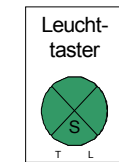


Bild 5: Bedienelemente (Beispiele)

Bild 6: Logikelemente (Beispiele)

des Verhalten einer Schaltschrank-Hardware modelliert: Als Reaktion auf einen kurzen Ausgangsimpuls der SPS steht am Schaltschrank nach einer Verzögerungszeit von ca. 500 msec bei positiver Rückmeldung der angeschlossenen Aggregate ein Signal zur Verfügung, das der SPS das erfolgreiche Anfahren dieser Aggregate quittiert.

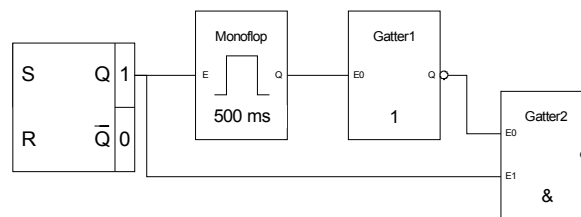


Bild 7: Modellierung einer „speichernden Einschaltverzögerung“

c) High-Level-Simulationsobjekte

Als High-Level-Simulationsobjekte bezeichnen wir solche Simulationsobjekte, deren Semantik sich aufgrund ihres hohen Abstraktionsniveaus und ihrer großen Komplexität nicht unmittelbar erschließt. Typische Vertreter dieser Kategorie sind das Achsmodul und das Sensormodul, die im folgenden Anwendungsbeispiel kurz erläutert werden.

5. Anwendungsbeispiel

Zur Veranschaulichung des IMS-Konzepts wird nun auf zwei Teilaspekte eine konkrete Maschinensteuerung eingegangen. Hierbei handelt es sich um den Einschaltvorgang und die Handsteuerung einer Nebenachse einer Sondermaschine.

Das Bedienpult der Simulation (Bild 8) entspricht dem Bedienpult der realen Maschine. Zum Einschalten der Maschine muss ein Schlüsselschalter umgelegt und zweimal hintereinander der Leuchttaster „Steuerspannung EIN“ betätigt werden. Für die zweite Betätigung des Leuchttasters ist ein durch das SPS-Programm kontrolliertes Zeitfenster einzuhalten, über das der Bediener mit einem Leuchtsignal unterrichtet wird. Zum Einschalten muss ferner die Rückmeldung eines sicheren Zustands durch die Schaltschrank-Hardware erfolgen.

Das Steuerungsprogramm der Maschine wird während der Simulation im Original benutzt, um die Modellelemente – hier insbesondere das Bedienpult – wie in der Realität zu steuern. Bild 9 zeigt den Zustand des Bedienpults nach erfolgreicher Durchführung der Einschaltpro-

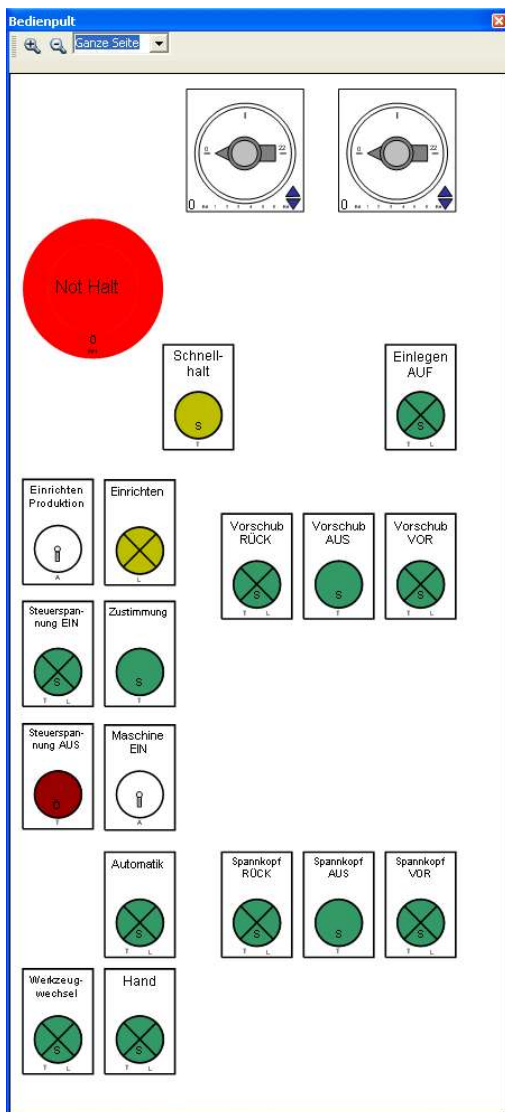


Bild 8: Bedienpult

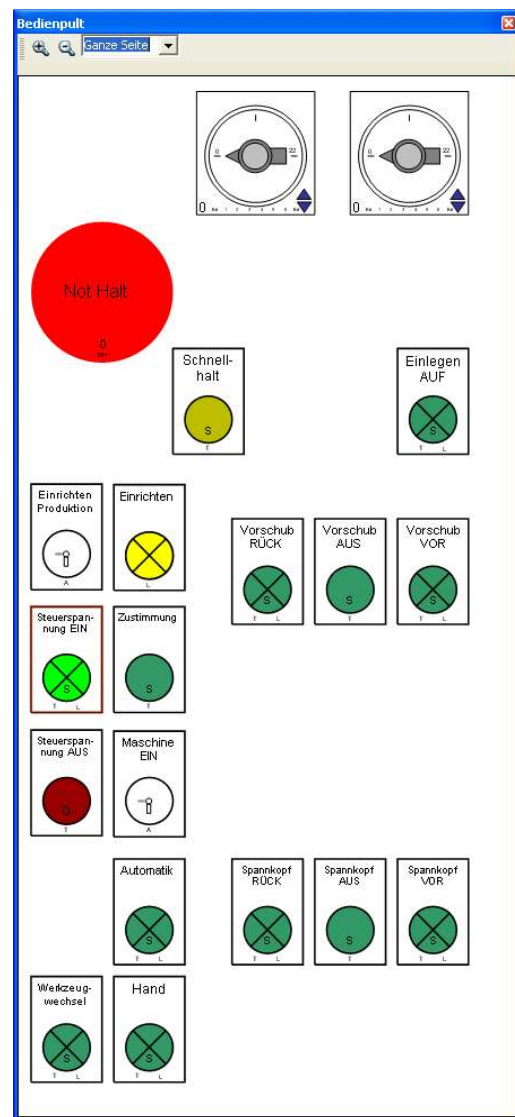


Bild 9: Bedienpult nach Einschaltprozedur

zedur. Man erkennt die veränderte Stellung der Schlüsselschalter sowie die Signalisierung der Leuchtmelder „Steuerspannung EIN“ und „Einrichten“.

In der Betriebsart „Einrichten“ können Achsen einzeln von Hand verfahren werden, wobei mit einem angepassten Sicherheitskonzept gearbeitet wird: Die Wirkung der in das SSI-Zählermodul der Achse integrierten Software-Endschalter sowie der Hardware-Endschalter, die durch zwei Sensorik-Module modelliert werden, kann verändert werden, um beispielsweise die Parametrierung nach einem Geberwechsel zu überprüfen. Erst mit Hilfe der Simulation können nun auch spezielle Fehlersituationen, wie etwa eine falsche Parametrierung, gefahrlos untersucht werden. Wie in Bild 10 dargestellt, wird das SSI-Zählermodul als Ursache eines fehlerhaften Achszustands nach einem Geberwechsel durch das Simulationswerkzeug verdeutlicht.

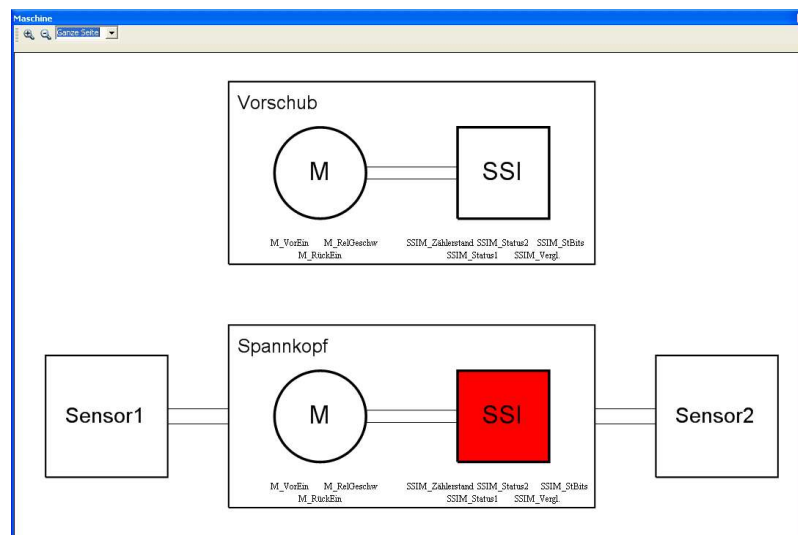


Bild 10: Achszustand nach falscher Geberparametrierung

Aus Sicht des Anwenders hat die Beschreibung der Simulationsobjekte und der zwischen ihnen bestehenden Verbindungen mittels XML zwei entscheidende Vorteile:

1. Er kann die bisher in seiner Prozesskette genutzten Werkzeuge auch für die Simulation nutzen. Für die Erstellung des Simulationsmodells sind nur wenige zusätzliche Prozessschritte erforderlich. Wie das Anwendungsbeispiel zeigt, müssen insbesondere Daten aus der Bedienfeld-Erstellung für die Simulation nicht erneut eingegeben werden. Diese Form der Integration in eine bestehende Prozesskette wäre ohne XML nicht möglich.
2. Durch die Beschreibung der Syntax des Simulationsmodells mittels XML macht der Anwender sich gegenüber dem Anbieter des Simulationswerkzeugs stärker unabhängig. Der Aufwand zur Nutzung des Modells mit einem anderen Simulationswerkzeug ist im Vergleich zu einem durch ein rein proprietäres Modellformat beschriebenes Modell deutlich geringer.

Aus unserer Sicht als Entwickler und Anbieter eines Simulationswerkzeugs bringt die Nutzung von XML zusätzlich folgende Vorteile:

1. Mit der Verfügbarkeit von „XML-Tools“ wie Schema-Editoren oder Parsern steht für die eigene Applikationsentwicklung eine gute Hilfestellung zur Verfügung.
2. Die Lesbarkeit von XML erleichtert das Debugging während der Entwicklung.
3. XML steht für Offenheit und ist somit auch ein Marketing-Argument.

6. Zusammenfassung

Die Wiederverwendung mechatronischer Komponenten im Maschinenbau kann durch die Simulation der Steuerung und Bedienung des neuen Gesamtsystems entscheidend unterstützt werden. Aus Anwendersicht ist hierbei die Integration des Simulationssystems in die bestehende Prozesskette von entscheidender Bedeutung. Diese Integration wird durch die Nutzung von XML im Vergleich zu einem Aufsetzen auf proprietäre Lösungen deutlich vereinfacht. Die Verschaltung automatisierungstechnischer Simulationsobjekte, die zumindest teilweise aus der bestehenden Engineering-Prozesskette übernommen werden, erlaubt dem Anwender die simulative Prüfung einer durch Wiederverwendung realisierten neuen Maschine. Die XML-Beschreibung des hierzu benutzten Simulationsmodells bietet neben dem Aspekt der Prozessintegration auch ein Potenzial für zukünftige offene Simulationssysteme.

Literatur

- [1] Albrecht, H.; Meyer, D.: XML in der Automatisierungstechnik – Babylon des Informationsaustausches? at – Automatisierungstechnik 50, 87-96, Oldenbourg Verlag, 2002.
- [2] Angerbauer, R.: SPS-Software aus dem Baukasten – Ein Erfahrungsbericht. VDMA Nachrichten 01, 45-46, 2002.
- [3] Eigner, M.; Stelzer, R.: Produktdatenmanagement-Systeme – Ein Leitfaden für Product Development und Life Cycle Management. Springer-Verlag, 2001.
- [4] Welp, E.G.; Lippold, C.; Bludau, C.: Ein System zur objektorientierten Modellierung mechatronischer Produktkonzepte (ModCoDe). VDI Mechatronik Tagung 2001, VDI Berichte 1631, 93-112, VDI-Verlag, 2001.