

S. Karras¹, T. Pfeiffer¹, H.-M. Hanisch¹, V. Vyatkin²

¹Germany, Martin Luther University Halle-Wittenberg

²New Zealand, The University of Auckland

INTEGRATION MECHATRONISCHER OBJEKTE IN DIE VAIAS-SYSTEMARCHITEKTUR

Zusammenfassung. Ziel des Forschungsprojektes VAIAS (Akronym für *Validierbare Architekturen für industrielle Automatisierungssysteme*) ist die Erhöhung der Entwurfssicherheit und die Verbesserung der Rekonfigurierbarkeit von Produktionssystemen. In der präsentierten Arbeit steht die in die Architektur zu integrierende Flexibilität beim Steuerungsentwurf im Vordergrund. Der Einsatz leicht austauschbarer mechatronischer Komponenten soll eine einfache Rekonfiguration des Systems ermöglichen. Zunächst wurden angrenzende Arbeiten näher untersucht und auf ihre Tauglichkeit hin überprüft.

1 Einführung

Die Automatisierungssysteme der Fertigungsindustrie sind in ihrer Art mechatronisch, d.h. sie verbinden mechanische, elektrische und elektronische Merkmale. In letzter Zeit wurde eine Vielzahl von Arbeiten über mechatronische Architekturen und mechatronische Ontologien veröffentlicht. Was die Mehrheit dieser Veröffentlichungen gemeinsam hat, ist das Verständnis der heterogenen Natur des mechatronischen Wissens. Also bestand die Idee zur Entwicklung einer erfolgreichen Vorgehensweise zur Validierung von Automatisierungssystemen darin, auf vorhandene mechatronische Architekturen (und sie unterstützende Werkzeuge) aufzubauen und – wenn erforderlich – sie zu erweitern. Dieser Ansatz kann den Aufwand für die Handhabung geometrischer, hydraulischer, elektrischer und anderer Informationen verringern und dem Entwickler helfen, sich mehr auf die Fragen der Verhaltensmodellierung, der eingebetteten Steuerung, der Laufzeitenbetrachtung und der Simulation konzentrieren.

Ein weiteres spezifisches Merkmal der Herangehensweise bildet die klare Trennung von Modellen des ungesteuerten Verhaltens der mechatronischen Apparatur und ihrer Steuerung. Das Modell des Systemverhaltens ergibt sich aus der Anwendung der Steuerung auf das Modell der ungesteuerten Strecke. Dadurch besteht die Möglichkeit, verschiedene Steuerstrategien zu überprüfen oder zu simulieren.

Diese Kombination wird in VAIAS als Schlüsselfaktor für den Steuerungsentwurf angesehen, der schließlich zu einer schnelleren

Entwicklung besserer Systeme führen soll. Besonders hervorgehoben wird die Idee der eingebetteten Modellierung. Diese fügt sich nahtlos in das Konzept des Automatisierungsobjektes [1] ein.

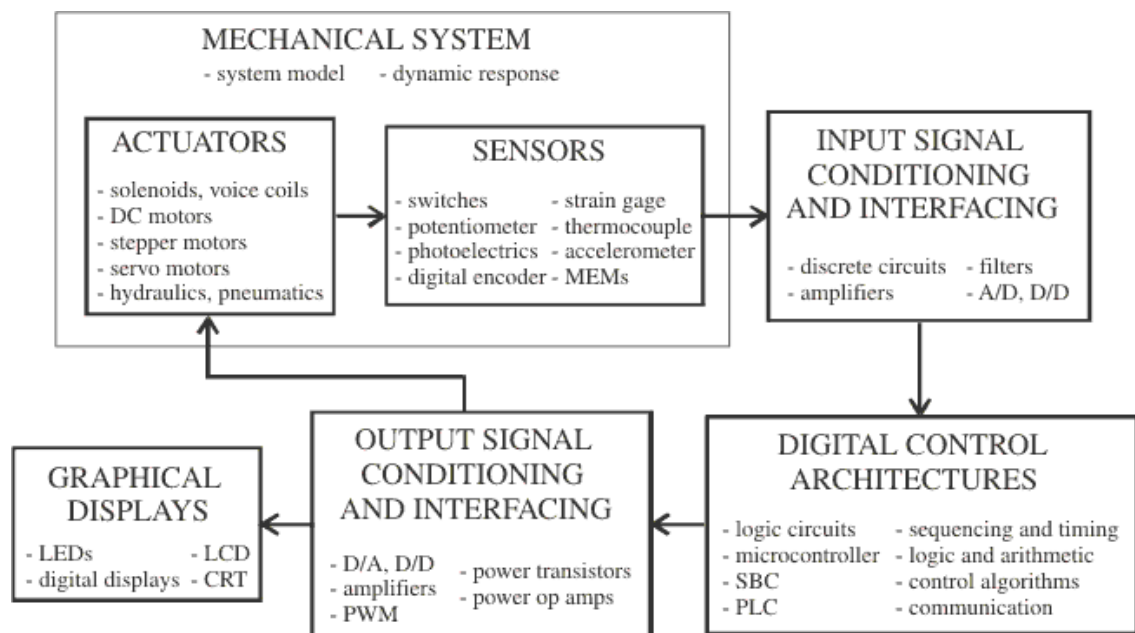


Abbildung 1. Allgemeine Struktur eines mechatronischen Systems [12]

Das Dokument ist wie folgt strukturiert: In Abschnitt 2 wird eine kurze Übersicht der letzten relevanten Arbeiten präsentiert, die sich mit mechatronischen Komponenten beschäftigen und für eine Eingliederung in VAIAS mit in Betracht kommen. Abschnitt 3 enthält eine Erörterung der VAIAS-Systemarchitektur, und die definierte Struktur einer mechatronischen Komponente wird anhand eines Beispiels illustriert. Aspekte der formalen Verifikation werden hier nicht berührt. Im Fokus stehen hier primär die beschreibenden Merkmale der konzipierten Architektur.

2. Definition mechatronischer Systeme zur Integration in die VAIAS-Systemarchitektur

In einem mechatronischen System wird das Zusammenwirken seiner mechanischen, elektronischen und informationstechnischen Bestandteile betrachtet [2]. Das Kunstwort Mechatronik (*Mechanical Engineering* / *Electronic Engineering*) ist ursprünglich ein Begriff aus der Feinmechanik, der etwa ab 1969 von der japanischen Firma *Yaskawa Electric Cooperation* geprägt wurde.

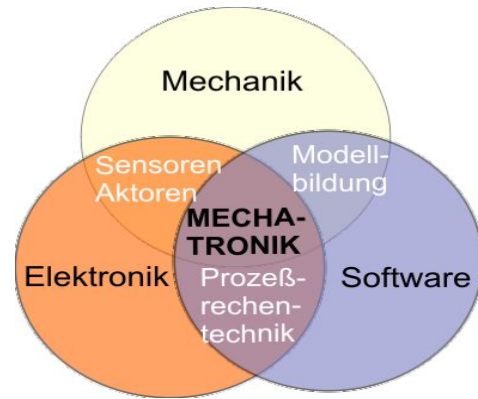


Abbildung 2. Mechatronisches System

Aus automatisierungstechnischer Sicht erscheint es sinnvoll, ein mechatronisches System in Hardware, Software und die zu deren Interaktion notwendigen Schnittstellen einzuteilen. Die Hardware (Aktoren, Sensoren, SPS-Gerät) umfasst sowohl mechanische, als auch elektronische Komponenten, die oft auch eine Einheit bilden. Der modifizierbare Teil des Steuerungs- und Informationssystems stellt die Software dar.

Betrachtet man ein mechatronisches System genauer, so wird man feststellen, dass das System für sich gesehen zwar eine gut strukturierte Einheit bildet, es jedoch weitergehender theoretischer Untersuchungen bedarf, um diese Einheit (Objekt) mit anderen nach dem Plug&Play-Prinzip zu verbinden bzw. in ein vorhandenes System zu integrieren.

In [6] und [14] werden mechatronische Systeme als eine Integration von vorwiegend mechanischen und elektrischen Systemen sowie einer dazugehörigen Informationsverarbeitung vorgestellt. Dieser Beitrag folgt dieser Darstellung, möchte jedoch einen noch umfassenderen Begriff prägen, um den Anforderungen an Objektorientiertheit und starker informationstechnischer Durchdringung Rechnung zu tragen.

Auf internationaler Ebene wurden bereits einige Anstrengungen unternommen, um u.a. mechatronische Produkte zu klassifizieren und zu katalogisieren. Ein Ergebnis ist die IEC 61360, welche technische Produkte anhand von Bauteilen, Materialien und Geometrien katalogisiert. Es wurden hierarchische Klassifikationen von Komponenten definiert sowie Datenelementtypen (DET), welche die Komponenten einer Klasse vollständig beschreiben. Die Norm zeigt deutlich, dass für rechnergestützte Entwicklungs- und Validierungsprozesse eine firmenübergreifende Standardisierung signifikanter Merkmale und Eigenschaften von der Industrie benötigt und auch gefordert wird.

Angestrebt wird in dieser Arbeit daher die Integration und technische Unterstützung von Konzepten für komplexe Intelligente Mechatronische Objekte (IMO), bei dem vorrangig die Funktionalität von Standardsoftware zum Einsatz kommt, um die Schwelle zum praktischen Einsatz möglichst tief zu halten. Dabei wird die Entwicklung problemspezifischer Software auf das absolut notwendigste reduziert und soll sich vorrangig auf die Entwicklung von Transformationswerkzeugen zur Konvertierung unterschiedlicher Datenformate beschränken.

Die objektorientierte Beschreibung von IMO-Komponenten wurde im Rahmen des bayrischen Forschungsprojektes „Mechasoft“ unter Leitung der TU München untersucht. Ein Ergebnis aus diesem Projekt ist ein Metamodell zur Beschreibung mechatronischer Komponenten (siehe Abbildung 3).

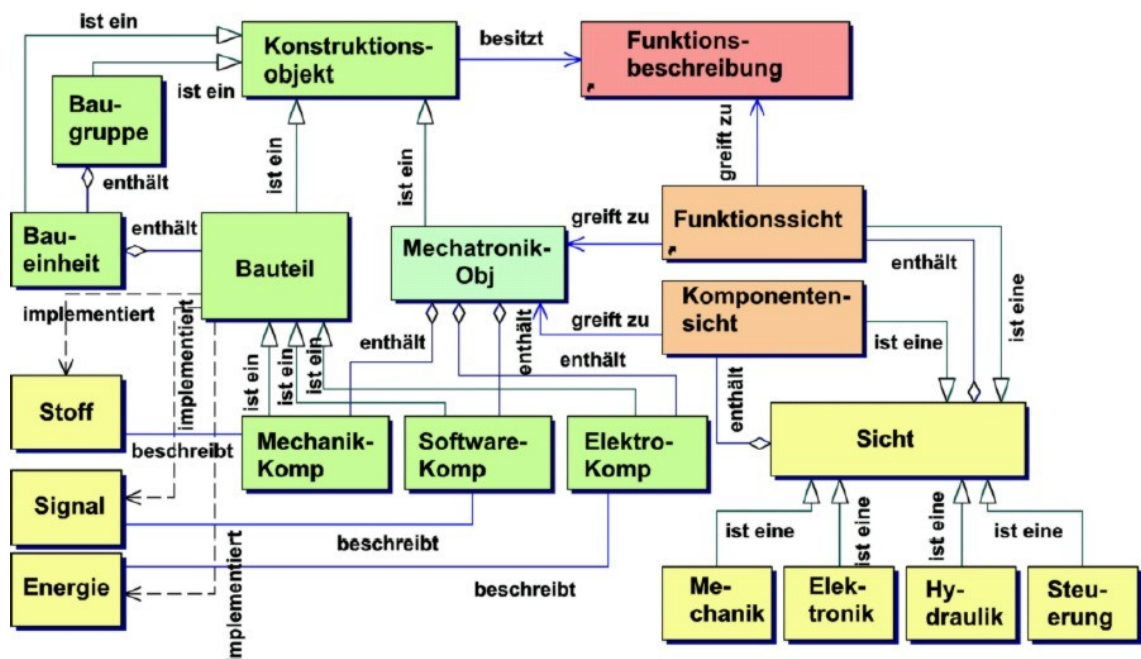


Abbildung 3. Metamodell einer mechatronischen Komponente[7]

Das Hauptaugenmerk des Mechasoft-Projektes [7] war auf die integrierte Entwicklung hochverfügbarer mechatronischer Systeme gerichtet. In [10] wurde eine eigens dafür geschaffene Entwicklungsumgebung zur Beschreibung und zum Aufbau von Simulationsmodellen vorgestellt.

Auf diese Arbeit wird aufgesetzt, um so eine objektorientierte Komposition der Komponenten zu erreichen. Dabei muss das Metamodell soweit angepasst werden, dass es – unabhängig von einer spezifischen

Branche – auch den allgemeinen Ansprüchen der Automatisierung genügt. Vor allem soll die intuitive Entwicklung und Komposition sowie die modellbasierte Verifikation von Steuerungsalgorithmen im geschlossenen Kreis unterstützt werden.

Bei der Beschreibung der mechatronischen Komponenten und deren Datenstruktur werden Notationen der Unified Modelling Language (UML) eingesetzt. UML wird ein großes Potential bei der Visualisierung, Entwicklung und Dokumentation von großen und komplexen mechatronischen Systemen zugesprochen [8,9], durch die sowohl der Entwicklungsprozess als auch die Komponentenstruktur anschaulich dargestellt werden. Ein methodologisches System für den Einsatz formaler Entwurfs- und Verifikationstechniken wird in [4,5] präsentiert. Der objektorientierte Modellierungsansatz basiert auf einer erweiterten Untermenge von UML, wobei verschiedene Diagramtypen der Modellierungssprache verwendet werden: Zunächst wird das System in der Top-Down-Vorgehensweise zerlegt in mechatronische Objekte, deren signalbezogene Schnittstellen von mechatronischen Klassen vorgegeben sind. Klassendiagramme, die das Strukturmodell abbilden, werden dann zur Beschreibung des hierarchischen Aufbaus des Systems verwendet. Dabei werden mechatronische Klassen über Composition-Links miteinander verbunden.

Dieser Ansatz bietet eine gute Unterstützung für die Entwurfsphase des Entwicklungsprozesses, da CASE Werkzeuge für die Modellspezifikation eingesetzt werden können und das dabei verwendete CORFU FBDK [15] eine automatische Modelltransformation gestattet. Zur Zeit fehlen jedoch formale Verifikationsmöglichkeiten. Dies könnte gelöst werden, wenn der in Bearbeitung befindliche Export von Funktionsblock-Netzwerken in IEC61499 konforme XML-Notation bereitsteht.

In den zuvor genannten Veröffentlichungen werden Wege aufgezeigt, die Struktur mechatronischer Objekte mit Hilfe entsprechender Metamodelle oder Ontologien zu beschreiben. Diese dienen uns als Grundlage für die Integration mechatronischer Objekte in die VAIAS-Systemarchitektur, deren Entwicklung im folgenden Abschnitt näher beschrieben werden soll.

3. Entwicklung der VAIAS - Systemarchitektur

Auf Basis der dargestellten Vorbetrachtungen kapselt ein IMO die für eine mechatronische Einheit maßgeblichen Daten und Softwarekomponenten. Das sind Implementierungen verschiedener

Funktionen, wie Ablauf- und Sicherheitssteuerung, Diagnose, Visualisierung etc. Sowohl für komplexe als auch für einfache IMO's wird von einer Schichtenstruktur ausgegangen, wobei eine Schicht über ihre Funktionalität definiert wird (z.B. Steuerung, Schnittstelle, Modell, Layout, Sicht usw.). Abbildung 4 illustriert dies anhand des einfachen IMO's „Linearantrieb“.

Eine Ebene kann Elemente verschiedenen Typs aufnehmen. Zum Beispiel könnte die Modellschicht ein generisches Modell in Form eines hybriden Zustandsdiagramms, dessen Programmimplementierung als Quellcode und ein vereinfachtes diskretes Modell als Petrinetz enthalten. Die Modellbildung wird als wesentlicher Bestandteil des Entwurfsprozesses betrachtet.

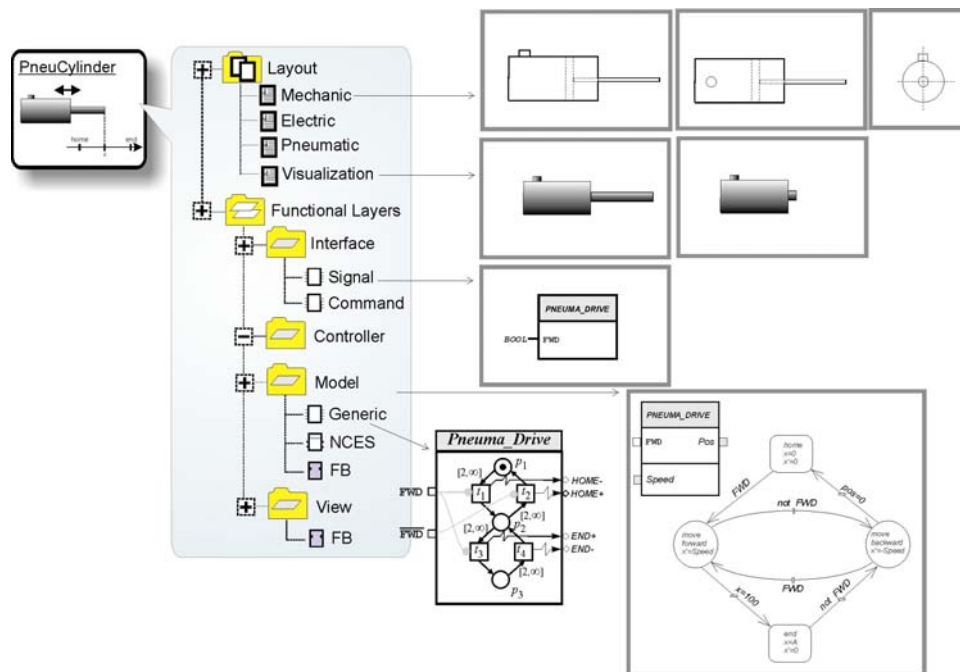


Abbildung 4. Schichtenstruktur eines einfachen mechanischen Objects

Als Referenz für die Ausführungsschicht wird die komponentenbasierte Architektur für Industrieautomatisierungssysteme des Standards IEC611499 dienen [13]. Das ermöglicht die Definition geräteunabhängiger verteilter Anwendungen und ihre Abbildung auf unterschiedliche Topologien von Hardwarebausteinen, die wiederum verschiedene Laufzeitplattformen haben können. Die Modelle werden die Simulation mit einstellbarer Genauigkeit, formale Verifikation und die Entwicklung oder Generierung von Code erlauben.

Eine exaktere Repräsentation von realen Objekten ergibt sich aus der Verfeinerung des Modells, bei der das IMO in noch kleinere Komponenten zerlegt wird, die wiederum mechatronische Objekte darstellen. Damit erhält man zusammengesetzte IMO. Andersherum - bei einem Bottom-up-Ansatz - können natürlich auch vorhandene einfache Objekte zu komplexeren zusammengesetzt werden [17,18].

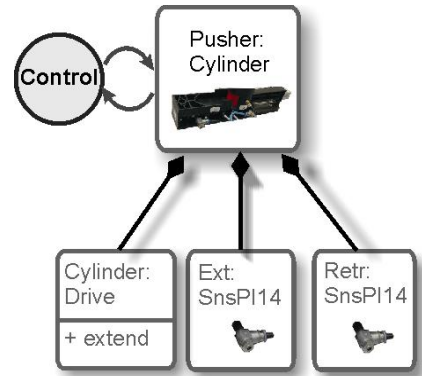


Abbildung 5. Zusammengesetztes mechatronisches Objekt

Abbildung 5 zeigt beispielhaft die hierarchische Struktur eines zusammengesetzten mechatronischen Objektes vom Typ „Zylinder“. Komplexe IMOs können auch aus anderen komplexen Objekten aufgebaut werden. Einfache IMOs bilden die Blätter der Baumstruktur und beinhalten keine weiteren Objekte.

Neben der objektorientierten Systembeschreibung findet auch die objektorientierte Softwareentwicklung in der Automatisierungstechnik zunehmend Anerkennung. VAIAS folgt diesem Ansatz. Der Entwurfsprozeß der Steuerung, basierend auf UML, besteht aus mehreren Schritten, die zu einer Verfeinerung informaler Anforderungen führen, bis diese die formale Gestalt ausführbaren Codes angenommen haben, der die Spezifikationen erfüllt (siehe Abbildung 6).

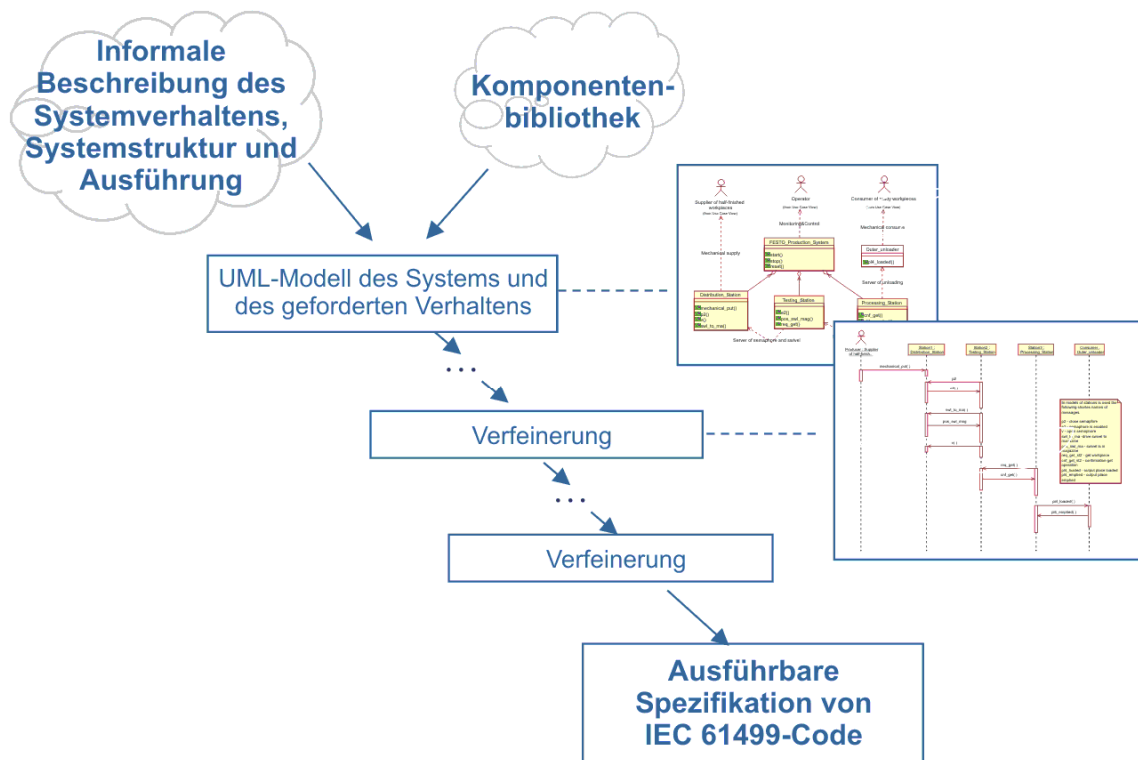


Abbildung 6. Steuerungsentwurf – ein zentraler Bestandteil der VAIAS-Systemarchitektur

Für die Strukturierung und Formalisierung der Anforderungen werden, ähnlich dem klassischen UML-Ansatz, verschiedene graphische Notationsformen verwendet. Auch Arbeiten wie [15,20] widmen sich diesem Gebiet und stellen unterschiedliche Ansätze vor. Der ausführbare Steuerungscode entspricht der IEC61499, womit die geforderte Flexibilität in der Verteilung und Rekonfigurierbarkeit erreicht werden soll [11,16]. Des Weiteren soll durch Anwendung des Standards die Zusammenarbeit von IEC61499-konformen Geräten verschiedener Hersteller realisiert werden.

VAIAS geht über den Ansatz des bloßen Steuerungsentwurfs hinaus und bindet zusätzlich Aspekte, wie die Integration der Anlage als eine Anordnung mechatronischer Komponenten und – dies ist als Schlüsselfaktor zu sehen – die Modellierung für die formale Verifikation als Teil des Systementwurfs im geschlossenen Kreis mit ein [19].

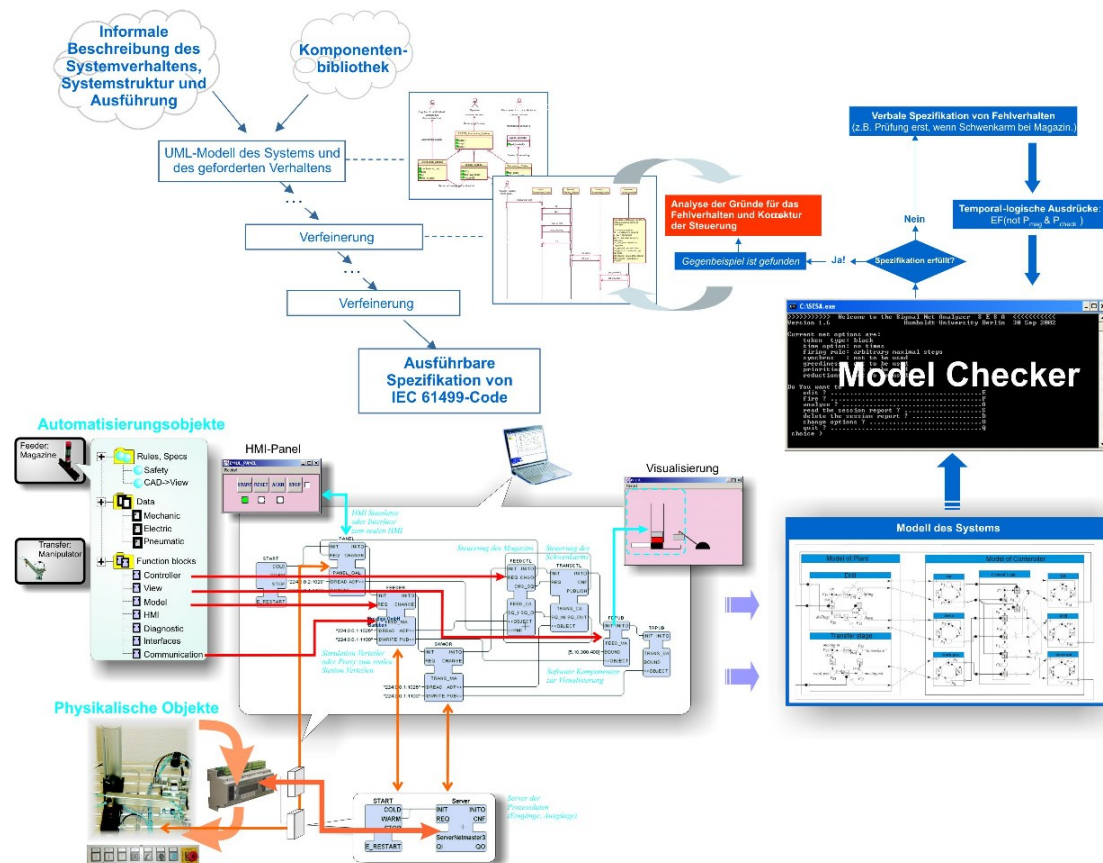


Abbildung 7. VAIAS - Systemarchitektur

Einen bedeutenden Bestandteil des Forschungsprojektes VAIAS bildet neben der Architektur (Abbildung 7) auch die Entwicklungsumgebung. Diese beinhaltet Komponenten wie Modell- und Codegeneratoren und Validierungswerkzeuge. Die VAIAS Entwicklungsumgebung ist flexibel gestaltet und wird in jedem Arbeitsschritt auch die manuelle Interaktion erlauben.

Eine Auswahl verschiedener Werkzeuge und ihre Einordnung in Anwendungsbereiche ist in Abbildung 8 dargestellt, dabei geben die Beschriftungen an den Pfeilen die für den Datenaustausch verwendeten Dateiformate an. Der Modellaustausch mit und zwischen verschiedenen UML CASE-Werkzeugen, der mittels XMI (XML Metadata Interchange) erfolgt, wird derzeit untersucht und ist zum Teil implementiert.

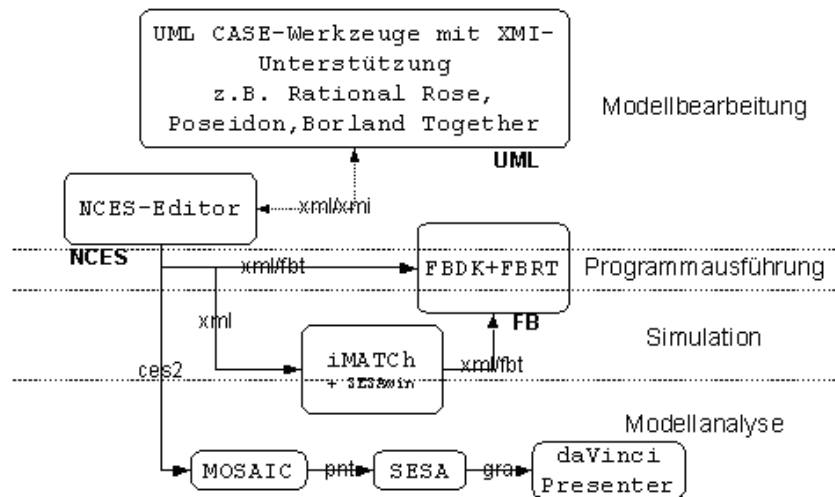


Abbildung 8. Werkzeugeinordnung in der Entwicklungsumgebung

4. Ergebnisse

Ziel des Forschungsprojektes VAIAS ist die Weiterentwicklung von Softwarearchitekturen und die Implementierung von entsprechenden Werkzeugen zur breiteren Anwendung von formalen Methoden für die Validierung von Software in der Industrieautomatisierung.

In dieser Arbeit wurde ein Ansatz für eine validierbare Softwarearchitektur unter Einbeziehung mechatronischer Komponenten vorgestellt, deren Grundlage das Konzept des Automatisierungsobjektes mit eingebetteten Modellen bildet.

Von vielen Forschern wird vor allem die Anwendung von UML für die Entwicklung von Industrieautomatisierungssystemen als Schlüsseltechnologie gesehen. Mit der dargestellten Architektur versucht VAIAS diesem Bestreben Rechnung zu tragen und darüber hinaus einen Beitrag hin zu systematisch entwickelter, fehlerfreier Steuerungssoftware zu leisten.

Danksagung

Das Forschungsprojekt wird vom Bundesministerium für Bildung und Forschung (BMBF) gefördert und von den Industriepartnern Pro-Sign GmbH und ifak system GmbH unterstützt.

Referenzen

1. Automation Objects for industrial-process measurement and control systems - IEC SB3/TC 65, Working draft, 2002

2. Gomma, H. Designing Concurrent, Distribute, and Real-Time Applications with UML, Addison Wesley 2000
3. Bruce Powel Douglass, "Doing Hard Time: Developing real-time systems with UML, Objects, Frameworks and Patterns" ISBN 1999
4. M. Bonfe, C. Fantuzzi: Design and Verification of Mechatronic Object-Oriented Models for Industrial Control Systems, IEEE Conference ETFA'2003, Lisbon, 2003, Proceedings, vol. II, pp.253-260
5. Bonfe M., Fantuzzi C. An Application of Object-Oriented Modeling Tools to Design the Logic Control System of a Packaging Machine, 2nd international Conference on Industrial Informatics INDIN'04, 24-26 June 2004, Berlin, Germany
6. Isermann, R.: „Mechatronische Systeme“, Springer Verlag, ISBN: 3540647252, 1999, Berlin.
7. Reinhart, G.; Anton, O.; Lercher, B.: "Funktionsorientiertes Sichtenmodell für die Entwicklung mechatronischer Systeme". VDI-Z 143 (2001) 11/12, S. 67-70
8. Mrozek, Z: "UML as integration tool for design of the mechatronic system", in Second Workshop on Robot Motion and Control, pp 189-194, ed. Kozłowski K, Galicki M, Tchoń K, Oct 18-20, 2001, Bukowy Dworek, Poland.
9. Mrozek, Z.: "Design of the mechatronic system with help of UML diagrams", 3-rd Workshop on Robot Motion and Control, 2002, Bukowy Dworek, Poland.
10. Zäh, M. F., Lercher, B., Pörnbacher, C., Wunsch, G.: „Anforderungen an das Datenmanagement in der Mechatronik“, in Industriemanager, 3/2003, Berlin.
11. H.-M. Hanisch and V. Vyatkin: Achieving Reconfigurability of Automation Systems by Using the New International Standard IEC 61499: A Developer's View, The Industrial Information Technology Handbook, CRC Press, in print, to appear in November 2004
12. Alciatore, D.; Hestand, M.: "Introduction to Mechatronics and Measurement Systems", 2nd edition, McGraw-Hill, ISBN: 007-240241-5, 2003.
13. J.H. Christensen: Design patterns for system engineering with IEC 61499. Proc. of Conference "Verteilte Automatisierung" (Distributed Automation), pages 63--71, Magdeburg, Germany, 2000
14. K. Feldmann, W. Wolf, M. Weber: Development of an Open, Event-based and Platform Independent Architecture for Distributed and Intelligent Control Systems, INDIN'04, Proceedings, pp. 560-566

15. Thramboulidis K.S. Using UML in Control and Automation: A Model Driven Approach, 2nd international Conference on Industrial Informatics INDIN'04, 24-26 June 2004, Berlin, Germany
16. Cai X., Vyatkin V., Hanisch H.-M. Design and Implementation of a Prototype Control System According to IEC 61499, IEEE Conference on Emerging Technologies in Factory Automation (ETFA'03), Proceedings, Lisbon, Portugal, September 2003
17. Vyatkin V., J. LM Lastra: Architectural Foundations for Reconfigurable Manufacturing Systems, 3rd International Symposium on Open Control Systems SoftSympo'03, Helsinki, September, 2003
18. Vyatkin V.: Intelligent Mechatronic Components: Control System Engineering using an Open Distributed Architecture, IEEE Conference on Emerging Technologies in Factory Automation (ETFA'03), Proceedings, Lisbon, Portugal, September 2003
19. Vyatkin V., Hanisch H.-M.: Verification of Distributed Control Systems in Intelligent Manufacturing, Journal of Intelligent Manufacturing, special issue on Internet Based Modeling in Intelligent Manufacturing, vol.14, N.1, 2003, pp.123-136
20. Zhang W., Halang W. und Diedrich Ch., Specification of Function Block Applications with UML, International Conference on Robotics and Automation ICRA05, Barcelona 2005