T U M

I N S T I T U T F Ü R I N F O R M A T I K

Integrierte Beschreibung automatisierter Produktionsanlagen - eine Evaluierung praxisnaher Beschreibungstechniken

Robert Sandner, Michael Mauderer

TUM–I9922
Dezember 99

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-12-I9922-100/1.-FI
Alle Rechte vorbehalten
Nachdruck auch auszugsweise verboten

©1999

Druck: Institut für Informatik der
Technischen Universität München

Beschreibungstechniken für mechatronische Systeme: Eine vergleichende Fallstudie

Robert Sandner[1], Michael Mauderer[2]

[1]Institut für Informatik
Technische Universität München
Arcisstraße 21, 80333 München
Email: Robert.Sandner@in.tum.de

[2]Institut für Werkzeugmaschinen und Betriebswissenschaften
Technische Universität München
Boltzmannstraße 15, 85748 Garching
Email: michael.mauderer@iwb.tum.de

Kurzfassung:
In einer vergleichenden Fallstudie werden drei Modellierungssprachen aus den Anwendungsbereichen des Maschinenbau und Software Entwicklung auf ihre Eignung für den Einsatz in einer integrierten Systementwicklung untersucht. Zum Einsatz kommen die Sprache EXPRESS, die dem von der ISO standardisierten STEP-Produktdatenmodell zugrunde liegt, die Sprache Real-Time Object-Oriented Modeling (ROOM) und die Unified Modeling Language (UML). Alle drei Sprachen werden zur Modellierung ausgewählter Teile einer am iwb aufgebauten Transferstraße eingesetzt. Im vorliegenden Bericht werden die Erfahrungen der Fallstudie zusammengefaßt. Betrachtet werden sowohl die Ausdrucksmächtigkeit und Strukturierungsmittel der Sprachen selbst als auch die Qualität der angebotenen Werkzeugunterstützung.

Dieser Bericht entstand im Rahmen des Projektes FORSOFT A4, unterstützt von der Bayerischen Forschungsstiftung und der Siemens AG.

Inhaltsverzeichnis

Inhaltsverzeichnis 2
1 Einführung 3
2 Durchführung der Fallstudie 4
2.1 Vergleichskriterien 4
2.2 Beschreibung des zu modellierenden Systems 5
2.3 Durchführung der Fallstudie 6
3 Beschreibung der eingesetzten Sprachen und Werkzeuge 7
3.1 UML 7
3.2 Real-Time Object-Oriented Modeling (ROOM) 8
3.3 EXPRESS 8
4 Ergebnisse 9
4.1 Ausdrucksmächtigkeit der Sprachen 10
Integrierte Modellierung von Hardware- und Softwarestrukturen 10
Integrierte Modellierung von Abläufen in Hardware- und Software 10
Strukturierung der Modelle 11
Zusammenwirken der Teilsprachen 12
4.2 Bewertung der Werkzeugunterstützung 12
Vollständigkeit des durch Werkzeuge unterstützten Sprachumfangs 12
Eignung der Werkzeugunterstützung zur Dokumentation 12
Benutzungsfreundlichkeit der Werkzeuge bei der Modellierung 13
Infrastruktur an Modellierungshilfen und Bibliotheken 13
4.3 Potential für Integration der drei Sprachen 14
5 Zusammenfassung und Diskussion der Ergebnisse 14
6 Literatur 16

1 Einführung

Die Entwicklung mechatronischer Systeme ist geprägt durch die Zusammenarbeit einer Vielzahl von Experten aus unterschiedlichen Fachrichtungen. In der Praxis sind die einzelnen Fachrichtungen häufig ungenügend in den Entwicklungsprozeß integriert – insbesondere ergeben sich vielfältige Brüche in der Informationsverarbeitung. Damit bleiben viele Einsparpotentiale ungenutzt: Viele Informationen im Entwicklungsprozeß werden an verschiedenen Stellen neu erarbeitet, sind inkonsistent oder nicht verfügbar, wenn sie benötigt werden. Die Folge sind zeitliche Verzögerungen und Mehrkosten in der Entwicklung. Ein wesentlicher Grund ist, daß die betroffenen Disziplinen im Laufe der Zeit unabhängig voneinander spezialisierte Vorgehensweisen und Beschreibungstechniken entwickelt haben, die nur wenig oder gar nicht integriert sind.

Um Einsparpotentiale im Entwicklungsprozeß zu eröffnen, ist dagegen als Basis ein integriertes Informationsmodell verbunden mit integrierten Beschreibungstechniken, über die alle Beteiligten das benötigte Wissen reibungslos austauschen können, erforderlich. Um eine integrierte Modellbildung zu ermöglichen, sind eine Reihe von Ansätzen denkbar. Vorstellbar ist es neue, auf die Anforderungen der Mechatronik hin optimierte, Modellierungssprachen zu entwickeln, um im Bereich der Mechatronik eine optimal integrierte Modellierung zu erreichen. Dies würde jedoch den Nachteil mit sich bringen, daß viele erprobte Methoden und Werkzeuge der einzelnen Disziplinen nicht mehr genutzt werden können. Daher ist es sinnvoll, existierende Sprachen in einen integrierten Ansatz einzubeziehen und möglichst eng zu integrieren. Wesentlich für die Nutzung von Synergieeffekten ist es, die Möglichkeiten des fachübergreifenden Einsatz der Techniken in einen solchen Ansatz einzubeziehen.

Ziel der im vorliegenden Bericht beschriebenen Fallstudie war es, die Eignung maßgeblicher Beschreibungstechniken aus den Disziplinen der Software Entwicklung und des Maschinenbaus im Hinblick auf eine integrierte Modellierung mechatronischer Systeme zu untersuchen. Besondere Aufmerksamkeit wurde der Möglichkeit des fachübergreifenden Einsatzes der Techniken gewidmet. In der Fallstudie wurden die folgenden Sprachen untersucht: Die Sprache EXPRESS [ISO94b] ist im Maschinenbau als Teil des STEP-Standards [ISO94a] die vorherrschende Beschreibungssprache für einen durchgehenden Informationsaustausch. In der Software Entwicklung hat die Unified Modeling Language [UML97a, UML97b, UML97c] besonders bei der Entwicklung betrieblicher Informationssysteme eine vorherrschende Bedeutung erlangt. Im Rahmen der maschinennahen Software Entwicklung werden jedoch in hohem Maße spezialisierte Techniken eingesetzt [WM85, SGW94]. In der jüngeren Vergangenheit hat vor allem die Sprache ROOM [SGW94] erhebliche Bedeutung erlangt, die ebenfalls in der Fallstudie betrachtet wird.

Für eine fundierte Aussage über die Praxiseignung der drei Sprachen zur Modellierung mechatronischer Systeme ist es unerläßlich, den Vergleich weitestgehend unter den Einsatz bedingungen der Praxis durchzuführen. Dazu wurde mit allen Sprachen eine real existierende, am Institut für Werkzeugmaschinen und Betriebswissenschaften der TU München aufgebaute Transferstraße soweit wie es die Sprachen ermöglichten modelliert. Die Modellierung wurde von drei Informatikstudenten mit Nebenfach Maschinenbau der TU München im Rahmen von Studienarbeiten durchgeführt. Die Bearbeitung der Fallstudie durch die Studenten ermöglichte einen weitgehend unvoreingenommenen Vergleich der Sprachen, da keiner der Studenten durch intensivere Erfahrungen mit einer der drei Sprachen „vorbelastet“ war. Die Erfahrungen in der Modellierung wurden von den Studenten in Ausarbeitungen und in Gruppendiskussionen mit den Betreuern verglichen.

Der vorliegende technische Bericht faßt die Erfahrungen der Fallstudie zusammen. Eine detailliertere Diskussion der Erfahrungen, die praktische Modellbeispiele einschließt, findet sich in den Ausarbeitungen der Studenten [Hof98,Kac98,Ler98]. Die Beschreibung der Fallstudie ist wie folgt gegliedert: In Abschnitt 2 werden der Aufbau der Fallstudie sowie die Vergleichskriterien, die an die Sprachen angelegt wurden, vorgestellt. Abschnitt 3 gibt eine Einführung in die eingesetzten Modellierungssprachen sowie die verwendeten Werkzeuge. In Abschnitt 4 werden die Erfahrungen der Fallstudie besprochen. Abschließend werden die Erfahrungen in Abschnitt 5 diskutiert und ein Resümee im

Hinblick auf den Einsatz der Sprachen in der integrierten Modellierung mechatronischer Systeme gezogen.

2 Durchführung der Fallstudie

2.1 Vergleichskriterien

Ziel der Fallstudie war es, für die praktische Modellierung relevante Aussagen zur Eignung der Sprachen UML, ROOM und EXPRESS in der Modellierung mechatronischer Systeme zu erhalten. Von besonderem Interesse war die Frage, ob die Sprachen über ihren ursprünglichen Einsatzbereich hinaus fachübergreifend eingesetzt werden können. Untersucht wurde deshalb, in wie weit UML und ROOM zur Beschreibung mechanischer Aspekte genutzt werden können, und in welchem Maß EXPRESS-Modelle eine Grundlage für die Entwicklung von Softwaresystemen darstellen können. Ein zweiter Schwerpunkt bei der Evaluierung war, ob die für die Sprachen verfügbaren Werkzeuge in der Modellierung mechatronischer Systeme eingesetzt werden können. Schließlich wurden die Möglichkeiten einer Integration der Sprachen untersucht. Im einzelnen wurden anhand der folgenden Vergleichskriterien untersucht:

- **Ausdrucksmächtigkeit der Modellierungssprachen:** Von maßgeblichem Interesse war die Frage, welche Aspekte sowohl der Systemstruktur als auch der Abläufe im System mit den jeweiligen Sprachen adäquat modelliert werden können. Untersucht wurde darüber hinaus, ob die Strukturierungsmechanismen für den Einsatzbereich adäquat und ausreichend sind. Schließlich wurde betrachtet, wie gut verschiedene Sichten des Entwicklungsprozesses, die ggf. mit unterschiedlichen Teilsprachen dargestellt werden, integriert werden können.
- **Bewertung der Werkzeugunterstützung:** Ein wichtiges Kriterium war, ob die vielfältigen Features der Sprachen, die in der Modellierung mechatronischer Systeme benötigt werden, durch die verfügbaren Werkzeuge vollständig unterstützt werden. Ein weiteres Kriterium für die Nutzung der Modelle in einem integrierten Entwicklungsprozeß war die Möglichkeit Dokumentationen zu generieren, die auch in anderen Werkzeugen weiter verwendet werden können. Ebenfalls betrachtet wurde die Bedienungsfreundlichkeit der Werkzeuge sowie die notwendige Einarbeitung. Der Schwerpunkt der Betrachtung lag hier auf identifizierten Defiziten. Eine quantitative Bewertung wurde nicht vorgenommen. Aussagekräftige quantitative Resultate würden eine wesentlich größere Anzahl von Testnutzern erfordern. Im Hinblick auf den wirtschaftlichen Einsatz wurde schließlich untersucht, in welchem Maß bei der Modellierung bereits auf wiederverwendbare Standardbausteine zurückgegriffen werden kann.
- **Integration der drei Sprachen:** Hier wird in knapper Form diskutiert, welche Integrationsmöglichkeiten durch übereinstimmende Konzepte der Sprachen bzw. geeignete Schnittstellen ggf. eine gemeinsame Nutzung der Sprachen ermöglichen.

Neben den genannten Kriterien sind natürlich noch weitere Vergleichskriterien denkbar. Im folgenden soll deshalb auch angesprochen werden, welche Aspekte in der Fallstudie *nicht* betrachtet wurden. Eine erhebliche Bedeutung für die Systementwicklung hat selbstverständlich die Verknüpfung von Entwicklungssprachen mit einer ausgereiften Vorgehensmethodik. Für die integrierte Entwicklung mechatronischer Systeme haben sich jedoch generell noch keine etablierten Vorgehensmodelle entwickelt. Die Einbindung etablierter Vorgehensmodelle sowohl aus dem Maschinenbau als auch aus der Software Entwicklung wirft vielfältige Fragestellungen auf, die unabhängig von den gewählten Modellierungssprachen behandelt werden müssen. Dies ist Gegenstand aktiver Forschungsarbeiten – auch der FORSOFT Projekte A4 und D1 [BM99]. Im Vordergrund dieses Berichtes steht der Vergleich von Modellierungssprachen, aus diesem Grund werden Fragestellungen der Vorgehens-

methodik nicht vorrangig betrachtet.[1] Ein übliches Kriterium in Vergleichen von Modellierungssprachen ist der Vergleich der Größe von Modellen für eine Anwendung. Eine detaillierte Messung der Modellgröße wurde in der Fallstudie nicht durchgeführt. Aufgrund des ähnlichen Abstraktionsgrades ergeben sich keine wesentlichen Unterschiede. Resultate in diesem Bereich würden bei weitem von den Unterschieden im Hinblick auf Anwendungsbereiche – die intensiv betrachtet werden – überlagert.

2.2 Beschreibung des zu modellierenden Systems

Aussagekräftige Resultate einer vergleichenden Fallstudie hinsichtlich der Eignung von Beschreibungssprachen in der Praxis der Systementwicklung erfordern, praxisrelevante Problemstellungen unter realitätsnahen Bedingungen zu bearbeiten. Um einen fairen Vergleich der entwickelten Modelle zu ermöglichen, ist eine Problemstellung nötig, die sowohl Fragestellungen des klassischen Maschinenbaus als auch der Steuerungs- und damit der Softwaretechnik in hinreichendem Umfang einschließt. Eine Aufgabenstellung, die sowohl den Forderungen nach Praxisnähe genügt als auch alle relevanten Fragestellungen in einem adäquaten Umfang einschließt, stellt die Modellierung der im folgenden beschriebenen Transferstraße dar. Die Anlage dient der Fertigung und Montage von Gehäusedeckeln und ist am Institut für Werkzeugmaschinen und Betriebswissenschaften aufgebaut.

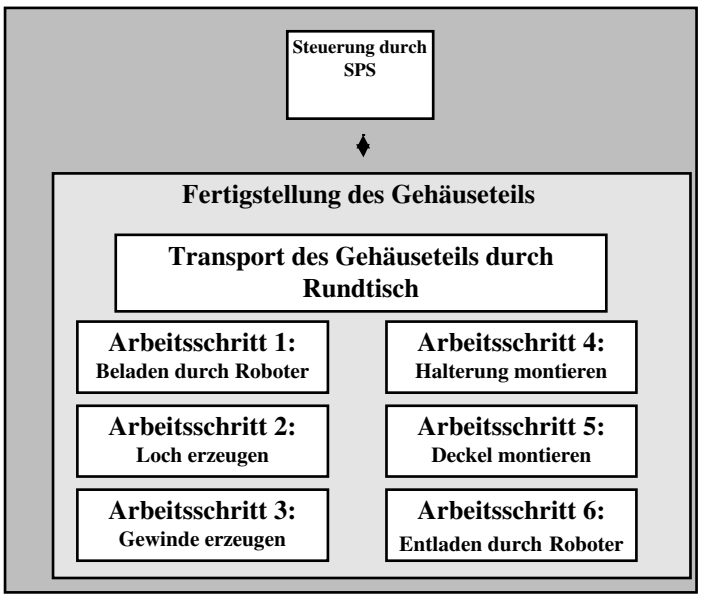

Abbildung 1: Teilaufgaben in der Transferstraße

Die Transferstraße hat die Aufgabe, einen Abschlußdeckel auf ein Aluminiumgehäuse zu montieren. Diese Produktion der Gehäuse umfaßt Fertigungs- und Montageaufgaben sowie Handhabungsvorgänge zum Be- und Entladen der Anlage. Die einzelnen Teilaufgaben zur Produktion der Gehäusedeckel werden in Abbildung 1 dargestellt. Nach dem Einbringen der Werkstücke in die Anlage werden zunächst in zwei Arbeitsgängen die Gewindebohrungen erzeugt. Anschließend wird der Gehäusedeckel eingepreßt und in einem weiteren Arbeitsgang mit dem Gehäuse verschraubt. Schließlich werden die Werkstücke durch einen Roboter wieder aus der Anlage entfernt. Jeder Arbeitsgang wird von einer separaten Bearbeitungsstation ausgeführt. Den Transport der Werkstücke zwischen den Stationen übernimmt ein sogenannter Rundtakttisch.

[1] Für den Bereich der Software Entwicklung existieren vergleichende Fallstudien, die methodische Fragestellungen einschließen. Hier sei insbesondere auf [KvK+99] verwiesen.

Sowohl die Konstruktion der Transferstraße als auch die Entwicklung der Steuerung stellen komplexe Entwicklungsaufgaben dar: Die mechanische Konstruktion erfordert, daß sechs Bearbeitungsstationen auf engem Raum angeordnet werden und über geeignete Schnittstellen verbunden werden. Die Aufgabe der Software Entwicklung ist insbesondere durch einen hohen Grad an Parallelität gekennzeichnet: Im Normalbetrieb werden sechs Werkstücke gleichzeitig in der Transferstraße bearbeitet, so daß eine Vielzahl von Abläufen koordiniert werden muß. Darüber hinaus muß die Steuerung eine Vielzahl unterschiedlicher Situationen kontrollieren können: Sowohl Teilbetrieb, Einrichtebetrieb als auch die Behandlung von Fehlersituationen – beispielsweise durch fehlerhafte Werkstücke – muß durch die Steuerung geleistet werden.

Die Steuerung der Anlage wird durch eine zentrale Steuerungshardware realisiert. In der Modellierung soll die Steuerungsaufgabe jedoch in adäquate Softwaremodule zerlegt werden. Obwohl die Steuerung als SPS-Steuerung realisiert ist, muß jedoch mit einer Vielzahl von Sensoren und Aktoren interagieren, wodurch auch Verteilungsaspekte berücksichtigt werden müssen. Einen Eindruck von der realen Anlage vermittelt Abbildung 2.

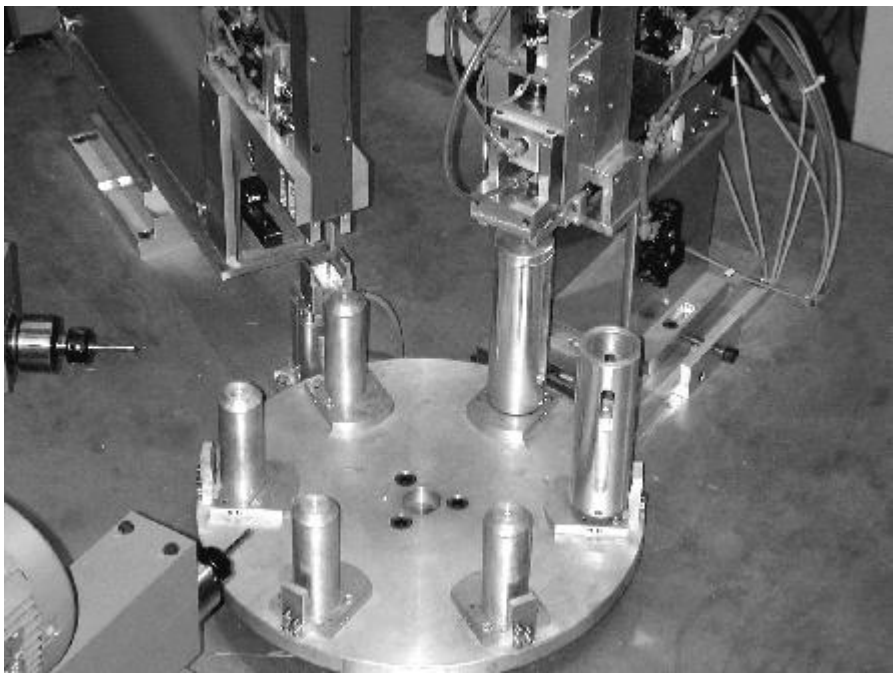

Abbildung 2: Teilefertigung in der zu modellierenden Transferstraße

2.3 Durchführung der Fallstudie

Um Erfahrungen mit hinsichtlich der Eignung der drei zu evaluierenden Sprachen zu sammeln, wurde die Transferstraße mit jeder der drei Sprachen modelliert. Ziel war es, sowohl ein möglichst breites Feld von Anwendungsbereichen abzudecken als auch die Vergleichbarkeit der Sprachen gewährleisten. Die Modellierung der Transferstraße umfaßt unterschiedlichste Anwendungsbereiche, so daß der ersten Forderung problemlos entsprochen werden konnte. Um die Vergleichbarkeit der Ergebnisse zu gewährleisten, wurde eine minimale Aufgabenstellung definiert, die mit allen drei Sprachen modelliert werden mußte. Diese Aufgabe umfaßte die Modellierung der Koordination der einzelnen Bearbeitungsstationen der Transferstraße sowie die detaillierte Modellierung eines Teilsystems. Hier wurde die Bearbeitungsstation ausgewählt, die die Gewindelöcher für den Abschlußdeckel bohrt.

Die Modellierung der Anlage wurde von Informatikstudenten der TU München im Rahmen eines sog. interdisziplinären Projektes durchgeführt. Alle Teilnehmer studierten das Nebenfach Maschinenbau

und konnten aus diesem Grund fundierte Kenntnisse für das Verständnis der Anwendung einbringen. Jeder der drei Studenten bekam die Aufgabe, mit je einer der Sprachen die Transferstraße gemäß der oben beschriebenen Aufgabenstellung zu modellieren. Die unterschiedlichen Modellierungen wurden teilweise in gemeinsamen Arbeitsgruppen von Betreuern und Studenten diskutiert. Damit sollte sichergestellt werden, Ideen der Projektteilnehmer zu transferieren und bei Diskrepanzen in der Modellbildung die Fähigkeiten der einzelnen Sprachen herauszuarbeiten.[2] Die Teilnehmer wurden angehalten, nicht nur das entwickelte Modell, sondern auch den Weg der Modellbildung zu dokumentieren und insbesondere Designentscheidungen, die durch die Modellierungssprache motiviert waren, zu erläutern. Bei der Erstellung der Modelle wurde auf Werkzeuge von Marktführern im jeweiligen Bereich zurückgegriffen. Für die Modellierung mit der UML wurde das Werkzeug Rational Rose verwendet, bei der ROOM-Modellierung kam das Werkzeug ObjecTime zum Einsatz, und für die Modellierung mit EXPRESS wurde der EXPRESS-Modeler der ProStep GmbH benutzt. Erfahrungen mit den Werkzeugen wurden von den Teilnehmern ebenfalls in Arbeitssitzungen mit den Betreuern diskutiert. Um subjektive Eindrücke zu vermeiden, wurde hier keine absolute Bewertung vorgenommen, sondern es wurden explizite, bei der Modellierung hinderliche Schwachpunkte aufgezeigt, diskutiert und in Ausarbeitungen dokumentiert.

3 Beschreibung der eingesetzten Sprachen und Werkzeuge

In diesem Abschnitt werden die in der Fallstudie untersuchten Modellierungssprachen sowie die eingesetzten Werkzeuge vorgestellt. Vertreten sind sowohl Sprachen aus der Software Entwicklung als auch aus dem Maschinenbau. Die Sprache EXPRESS [ISO94b] dient der Definition von standardisierten Produktmodellen, die im Bereich des Maschinenbaus den beliebigen Austausch von Daten zwischen verschiedenen Engineering Werkzeugen ermöglichen sollen. Die Unified Modeling Language [UML97a,UML97b,UML97c] wird vor allem bei der Entwicklung von betrieblichen Informationssystemen, die Sprache ROOM [SGW94] bei der Entwicklung echtzeitkritischer Softwaresysteme eingesetzt.

3.1 UML

Die *Unified Modeling Language* (UML) [UML97a, UML97a, UML97a] hat im Bereich der objektorientierten Entwicklung von Softwaresystemen eine führende Bedeutung erlangt. Vor allem bei der Entwicklung betrieblicher Informationssysteme hat die UML in der industriellen Praxis eine hohe Akzeptanz erlangt. Darüber hinaus existieren aber auch Ansätze, die UML auch für die Software Entwicklung in technischen Systemen einzusetzen [Dou97]. Die UML vereint, ausgehend von OOA&D [Boo94], OMT [RBP+91] und OOSE [Jac92] eine Reihe bedeutender Ansätze für objektorientierte Analyse und Design und wird seit 1997 von der OMG standardisiert.

Die vielfältigen Beschreibungstechniken der UML, deren Anwendungsbereiche sich teilweise überlappen, unterstützen vorwiegend die Analyse- und Designphase der Entwicklung, betrachten aber auch Aspekte der Implementierung. Für die Analysephase stellt die UML die Diagrammformen Use-Case Diagramm, Sequenzdiagramm und Zusammenspieldiagramm zur Verfügung. Use-Case Diagramme dienen der statischen Beschreibung von Anwendungsfällen. Sequenzdiagramme sind eine vereinfachte Variante von Message Sequence Charts (MSCs) [ITU96] und dienen wie Zusammenspieldiagramme der Darstellung von Interaktionen des Systems mit seiner Umgebung bzw. Interaktionen zwischen den Komponenten des Systems. Der konstruktive Systementwurf in der UML erfolgt mit Klassendiagrammen und Zustandsübergangsdiagrammen. Klassendiagramme stellen konzeptionelle Beziehungen zwischen Objektklassen, jedoch keine konkreten Systemstrukturen, dar. Eine

[2] Aus studientechnischen Gründen konnten nur zwei der Arbeiten zeitgleich durchgeführt werden. Die Modellierung mit der UML wurde bereits früher durchgeführt. Die Erfahrungen standen damit für die beiden anderen Projektteilnehmer vollständig zur Verfügung, während in umgekehrter Richtung nur teilweise Feedback erfolgen konnte. Trotzdem konnte aber erreicht werden, daß die Modelle fachlich weitgehend übereinstimmen.

Variante von Klassendiagrammen erlaubt zwar auch, Beziehungen zwischen konkreten Objekten darzustellen. Diese Darstellung wird jedoch kaum verwendet und von den Autoren der UML nur zur Darstellung von Beispielen für Datenstrukturen empfohlen [UML97b]. Die Zustandsübergangsdiagramme der UML sind eine leicht modifizierte Variante von Statecharts [Har87] und beschreiben das Verhalten von Objekten einer Klasse. Weitere Diagramme, die hier nicht explizit besprochen werden sollen, beschreiben Nebenläufigkeits- und Implementierungsaspekte.

Von den Diagrammarten der UML wurden in der Fallstudie vor allem die folgenden verwendet: Für die Beschreibung von Abläufen in der gesamten Transferstraße wurden die Diagrammarten Use-Case-Diagramm, Sequenzdiagramm und Zusammenspieldiagramm benutzt. Für die Beschreibung von Komponenten des Systems, insbesondere der Bohreinheit, wurden Zustandsübergangsdiagramme und Klassendiagramme verwendet.

Da die UML aus einer Reihe von akzeptierten Methoden hervorgeht und in der Praxis eine große Akzeptanz genießt, existiert eine breite Unterstützung durch Case Tools. Bekannte Tools sind *Rational Rose, Raphsody* , *Together* oder *Software trough Pictures*. Auch Werkzeuge, die ursprünglich für andere Modellierungssprachen, z.B. SDL [ITU93], entwickelt wurden, werden zum Teil auf die UML hin erweitert. Ein Beispiel für diese Entwicklung ist das SDL-Werkzeug ObjectGeode. In der Fallstudie wurde das Werkzeug Tool Rational Rose von der Fa. Rational Corp. verwendet, die derzeit Marktführer bei der Entwicklung von UML-Werkzeugen ist.

3.2 Real-Time Object-Oriented Modeling (ROOM)

Die Modellierungssprache und Methode Real-Time Object Oriented Modeling (ROOM) [SGW94] zielt speziell auf die Entwicklung von Telekommunikations- und Echtzeitsystemen ab. Im Vordergrund steht dabei die Unterstützung der Designphase des Entwicklungsprozesses sowie der Übergang zur Implementierung. Für dieses Anwendungsspektrum steht eine ausgewogene methodische Unterstützung zur Verfügung. Analog zur UML bietet ROOM eine Struktur- und Verhaltenssicht auf das System. Beide Sichten unterstützen durch eine ausgeprägte Schichtenbildung den schrittweisen Systementwurf.

Im Unterschied zur UML sind Klassendiagramme nicht Bestandteil der Modellierung in ROOM. Die Strukturbeschreibung des Systems erfolgt mit Aktordiagrammen, in denen der hierarchische Aufbau der Objekte (Aktoren) des Systems sowie deren Kommunikationsbeziehungen werden. Durch die Bildung von Gruppen gleichartiger Aktoren, deren Anzahl variieren kann, ist es möglich auch Mobilitätsaspekte zu modellieren. Mittels Ports werden eindeutige Schnittstellen zwischen Aktoren und ihrer Umgebung spezifiziert. Die Nachrichten, die zwischen Aktoren über Ports kommuniziert werden, werden mit Hilfe von Protokollen festgelegt. Protokolle werden unabhängig von Aktoren in Vererbungshierarchien beschrieben. Damit ist es möglich, die Entwicklung allgemeiner Kommunikationsarchitekturen von der Entwicklung konkreter Systeme abzutrennen.

Die Verhaltensbeschreibung der Aktoren des Systems erfolgt mit hierarchischen Zustandsmaschinen, beschrieben mit den sog. ROOM-Charts, die mittels asynchroner Kommunikation miteinander kommunizieren. Die Vergabe von Prioritäten für Nachrichten sowie Systemdienste wie Timer erlauben es auch, das Echtzeitverhalten der Aktoren zu beschreiben. Die Einbindung von der ROOM-eigenen Programmiersprache RPL sowie von C++ ermöglicht es, detaillierte Algorithmen in das Systemmodell mit einzubinden. Im Rahmen des kommerziellen Werkzeugs ObjecTime steht eine umfangreiche Unterstützung für die Ablaufsimulation entwickelter Modelle sowie für Code-Generierung zur Verfügung. Für die Dokumentation von Ablaufsimulationen ist schließlich die Generierung einer vereinfachten Variante von Message Sequence Charts in das Werkzeug integriert.

3.3 EXPRESS

Die Modellierungssprache EXPRESS [ISO94b] wurde im Rahmen von Bemühungen zur Vereinheitlichung von Produktdatenmodellen des Maschinenbaus entwickelt. Sie ist Grundlage des *Standard for the Exchange of Product Model Data (STEP)* [ISO94a]. Der Standard strebt, ausgehend vom

Austausch von Datenmodellen zwischen CAD-Systemen, eine umfassende Normung der Produktdaten des Maschinenbaus in Referenzmodellen an, um Ingenieuren den vollständigen Austausch aller in der Entwicklung eines Produkts benötigten Informationen unabhängig von den verwendeten Engineering-Werkzeugen zu ermöglichen. Die Weiterentwicklung des Standars wird von der ISO in der Normengruppe ISO-10303 vorangetrieben. Die in EXPRESS beschriebenen Referenzmodelle des Standards konzentrieren sich jeweils auf die Formalisierung von spezifischen Beschreibungsaspekten, die im Rahmen des Entwicklungsprozesses abgebildet werden müssen. Für diese Modelle werden von den Herstellern von Engineering Werkzeugen Schnittstellen implementiert, so daß über diese Modelle ein verlustfreier Datenaustausch möglich ist. Von besonderer Bedeutung für die integrierte Entwicklung mechatronischer Systeme sind die Teilnormen für die Formalisierung von Geometriebeschreibungen [ISO94d], Kinematikbeschreibungen [ISO94e], aber auch von Fertigungsschritten zur Herstellung eines Produkts [ISO94f]. Bislang wird in der industriellen Praxis nur ein Teil der STEP-Normen unterstützt. Im Rahmen der Weiterentwicklung des Standards ist jedoch ein zunehmender Einfluß auf weitere Anwendungsaspekte zu erwarten, da sich viele der Normen noch in der Entwicklung befinden und noch nicht abschließend von der ISO verabschiedet sind.

Die Sprache EXPRESS unterstützt ausschließlich die Entwicklung von Datenmodellen in der Systementwicklung. Im Unterschied zu UML und ROOM stehen keine Möglichkeiten für die Darstellung von Systemabläufen zur Verfügung. EXPRESS basiert auf objektorientierten Erweiterungen von Entitiy Relationship-Diagrammen [Chen76] und umfaßt sowohl eine textuelle als auch eine graphische Notation. Als Erweiterung der klassischen ER-Diagramme ermöglicht EXPRESS analog zu Klassendiagrammen die Darstellung von Vererbungsbeziehungen zwischen Entitäten. Über die in den meisten objektorientierten Sprachen üblichen Darstellungsmittel hinausgehend stehen auch Operatoren zur Verknüpfung von Subtypen[3] einer Vererbungsbeziehung zur Verfügung: Die Operatoren ANDOR bzw. AND zwischen Subtypen in einer Vererbungsbeziehung legen fest, daß deren Instanzen beide Subtypen ausfüllen können bzw. müssen. Darüber hinaus stellt EXPRESS eine umfangreiche Spezifikationssprache zur Verfügung, um logische bzw. algorithmische Beziehungen zwischen Entitäten, die im Relationenkalkül nicht ausgedrückt werden können, zu spezifizieren. Allerdings ist diese Sprache nicht in die graphische EXPRESS-Notation integriert. Zur Strukturierung der umfangreicher Modelle, wie sie die Normen des STEP-Standards darstellen, stehen feingranulare Mechanismen zur Modulbildung zur Verfügung. Damit ist es möglich, sowohl die Normen des Standards bei der Entwicklung eigener Modelle als Module zu importieren als auch eigene Modulbibliotheken zu erstellen.

Zur Entwicklung von EXPRESS-Modellen werden von verschiedenen Herstellern Werkzeuge angeboten. Im deutschsprachigen Raum hat der EXPRESS Modeler der ProSTEP-GmbH eine führende Stellung erlangt. Dieses Werkzeug wurde in der Fallstudie eingesetzt.

4 Ergebnisse

In diesem Abschnitt werden die Erfahrungen, die in der Fallstudie hinsichtlich der Anwendungsmöglichkeiten der drei Sprachen gesammelt wurden, vorgestellt. Entsprechend den in Abschnitt 2 besprochenen Zielsetzungen ist die Diskussion der Ergebnisse in drei Gruppen von Kriterien gegliedert: In Abschnitt 4.1 wird die Ausdrucksmächtigkeit der Sprachen untersucht. Dies umfaßt sowohl die Eignung für die Modellierung des mechanischen Hardwareanteils mechatronischer Systeme als auch die Modellierung von Abläufen und des Systemverhaltens. Ebenfalls betrachtet werden Strukturierungsmechanismen sowie die Integration der Teilsprachen, die jeweils in den drei Sprachen zur Verfügung stehen. In Abschnitt 4.2 werden die in der Fallstudie eingesetzten Werkzeuge kritisch betrachtet. Gegenstand der Diskussion sind die Vollständigkeit des unterstützten Sprachum-

[3] Im Unterschied zu klassischen objektorientierten Sprachen wird in EXPRESS von Entities bzw. Typen anstatt von Klassen gesprochen. Subtypen entsprechen demnach Subklassen in der Terminologie der Objektorientierung. Analog zur Objektorientierung werden konkrete Beispiele von Entitäten als Instanzen bezeichnet.

fangs, die angebotene Funktionalität zur Dokumentation erstellter Modelle, die Verständlichkeit der Bedienerführung sowie die Verfügbarkeit von Bibliotheken, auf die bei der Modellierung zurückgegriffen werden kann. Schließlich wird in Abschnitt 4.3 diskutiert, in welchem Maß eine gemeinsame Nutzung der Sprachen für die Systementwicklung sinnvoll und möglich ist.

4.1 Ausdrucksmächtigkeit der Sprachen

Integrierte Modellierung von Hardware- und Softwarestrukturen

Eine wesentliche Anforderung an Modellierungssprachen in der integrierten Entwicklung mechatronischer Systeme ist es, mechanische Hardware und Software möglichst aus einem gemeinsamen Modell heraus zu entwickeln. Allgemeine topologische Beziehungen können in allen drei Sprachen gut abgebildet werden. Als besonders gut geeignet haben sich ROOM-Aktordiagramme erwiesen. Sie ermöglichen sowohl eine intuitive Modellierung der Zerlegung des Gesamtsystems in einzelne Komponenten als auch die Abbildung der Kommunikations-Infrastruktor des Systems. Insbesondere bei der detaillierten Modellierung der Bohreinheit wurden mit ROOM sehr gute Erfahrungen gesammelt. Ansätze, die Modellierungskonzepte von ROOM auch in die UML zu integrieren, finden sich in [SR98]. Allerdings ist die methodische Integration dieser Konzepte in die UML noch nicht ausreichend geklärt.

In keiner der Sprachen ist es möglich, die in der Mechanikentwicklung eminent wichtigen geometrischen Systemstrukturen unmittelbar zu visualisieren. EXPRESS bietet die Möglichkeit, auf die in STEP definierten Geometriemodelle zurückzugreifen, die mittels CAD-Werkzeugen dargestellt werden können. Die Modellerstellung ausgehend von EXPRESS-Modellen ist jedoch wenig intuitiv und für komplexe Modelle ungeeignet. Ein geometrische Systementwurf kann nur ausgehend von mit CAD-Systemen erstellten Modellen erfolgen, die in ein Gesamtmodell integriert werden. Hierfür steht allerdings derzeit noch keine ausreichende Werkzeugunterstützung zur Verfügung, so daß diese Tragfähigkeit dieses Ansatzes nicht am praktischen Beispiel überprüft werden konnte.

In der methodischen Konstruktion wird in modernen Ansätzen ein Modellierungsansatz vorgeschlagen, der bei der Modellierung der Topologie zwischen Material-, Energie- und Informationsfluß unterscheidet [Rod91, PB93]. Diese Unterscheidung kann zwar in allen Sprachen durch die Definition geeigneter Datentypen für Kommunikationsstrukturen modelliert werden. Allerdings ist weder in ROOM und UML noch in EXPRESS eine Visualisierung der Darstellung vorgesehen. Prinzipiell ist es allerdings möglich, in der UML auf Basis der von der Sprache angebotenen Erweiterungsmechanismen eine solche Darstellung einzuführen.

Integrierte Modellierung von Abläufen in Hardware- und Software

Bei der Modellierung von Ablauf- und Verhaltensaspekten werden im folgenden nur die Sprachen UML und ROOM betrachtet, da EXPRESS, wie in Abschnitt 3 besprochen wurde, die Modellierung von Abläufen nicht unterstützt. Zu bemerken ist hier, daß auch im Maschinenbau verbreitete Ansätze nicht in EXPRESS integriert sind. Um zu ermitteln, ob eine Anbindung solcher Ansätze an EXPRESS hilfreich ist, wurden Zustandsdiagramme nach VDI 3260 [VDI77] in die Betrachtung mit einbezogen. Für die Darstellung des Zusammenwirkens des Mechanik- und Softwareanteils haben sich diese Diagramme prinzipiell als hilfreich erwiesen, da sie in begrenztem Maße auch kontinuierliche Zustandsänderungen von Systemkomponenten abbilden können. Allerdings werden die Diagramme bereits in den in der Fallstudie betrachteten Anwendungsfällen so umfangreich, daß ein Einsatz in großen Applikationen nicht sinnvoll erscheint.

Bei der Bewertung der Eignung von ROOM und UML zur Modellierung von Systemabläufen ist eine differenzierte Betrachtung der Abstraktionsebenen in der Anlagenentwicklung nötig. Auf der *Anlagen- und Zellen ebene,* die in der Fallstudie durch die Modellierung der Abläufe in der gesamten Transferstraße repräsentiert wird, muß vor allem der Informations- und Materialfluß zwischen den Bearbeitungs-, Materialfluß- und Informationssystemen betrachtet werden. Physikalische Aspekte sind hier von untergeordneter Bedeutung. Sowohl mit der UML als auch mit ROOM wurden dabei

gute Erfahrungen gesammelt. Bei der Modellierung der Interaktion der Komponenten der Transferstraße haben sich Sequenzdiagramme und Zusammenspieldiagramme der UML als hilfreich erwiesen. In ROOM ist zwar die Verwendung von MSCs vorgesehen, aber bislang nur zur Dokumentation der Simulation von Modellen integriert, so daß sie in ROOM nicht eingesetzt werden konnten. Bei der Modellierung der Anlagen- und Zellen ebene mit der UML wurden die folgenden Erfahrungen gesammelt: Die Verwendung von Sequenzdiagrammen führt bei der Darstellung der hochparallelen Abläufe in der Transferstraße zu extrem umfangreichen und damit unübersichtlichen Diagrammen. Eine Alternative für die übersichtliche Betrachtung von Teilabläufen, etwa die Bearbeitung eines einzelnen Werkstücks, stellen Zusammenspieldiagrammen dar. Allerdings ist es dabei nicht möglich, zeitliche Abhängigkeiten zwischen verschiedenen Teilabläufen, z.B. die Bearbeitung von zwei Werkstücken, zu betrachten.

Auf der *Maschinenebene,* die in der Fallstudie durch die Modellierung der Bohreinheit repräsentiert wird, eignen sich sowohl UML als auch ROOM nur bedingt für die integrierte Modellierung, da physikalische Effekte nur unzureichend abgebildet werden. Die Zuverlässigkeit des Systems hängt wesentlich von kontinuierlichen Zustandsänderungen – z.B. Bewegungen – und insbesondere von deren Güte ab. Das Zusammenwirken zwischen Softwaresystemen und mechanischen Abläufen bzw. deren Güte konnte nur unzureichend modelliert werden. Versuche, dieses Zusammenwirken zu modellieren, führten dazu, daß im wesentlichen die gleichen Abläufe für Software und Hardware modelliert wurden.[4]

Strukturierung der Modelle

Die Entwicklung mechatronischer Systeme ist stark von der Wiederverwendung existierender Teillösungen und der Zerlegung des Systems in Einzelkomponenten geprägt. Diese Zerlegung des Modells entspricht unmittelbar den Strukturierungsmechanismen der Aktordiagramme in ROOM. Allerdings führt die strikte Orientierung an der Zerlegung von Komponenten in Teilsysteme bei großen Modellen zu tiefen Verschachtelungen der Teilmodelle. Diese Schichtenbildung führt bei großen Anwendungen tendentiell zu unübersichtlichen Modellen, da jede Schicht als eigenes Teilmodell dargestellt wird und somit Abhängigkeiten zwischen einer Vielzahl von Modellen zu betrachten sind. Das Fehlen von Vererbungsbeziehungen zwischen Aktoren – Vererbungen sind nur zwischen Schnittstellen möglich – führt dazu, daß die Wiederverwendung von fertigen Teillösungen allerdings nur unzureichend unterstützt wird.

Die Strukturierung von Modellen anhand von Klassen in der UML bzw. von Entitäten in EXPRESS birgt dagegen die Gefahr, konzeptionelle Beziehungen zwischen Klassen mit der Darstellung konkreter Kommunikationsstrukturen zu vermengen. Eine Trennung dieser Aspekte wird in beiden Sprachen nur unzureichend unterstützt. Im Rahmen der Entwicklung einer Echtzeit-Variante der UML (UML-RT) werden Zusammenspieldiagramme erweitert, um diese Aspekte darzustellen. Allerdings sind diese Konzepte, wie bereits angesprochen, noch nicht ausreichend in die Sprache integriert. Für EXPRESS steht die Variante EXPRESS-I [ISO94c] zur Verfügung. Diese Sprache ist allerdings lediglich für die Entwicklung von Beispielstrukturen[5] konzipiert und ermöglicht keine graphische Visualisierung der definierten Strukturen.

Große Modelle lassen sich in UML bzw. EXPRESS mittels der zur Verfügung stehenden Mechanismen zur Modulbildung (Packages) bzw. Schemata) gut strukturieren. Die Entscheidung, welche Teilmodelle in Modulen zusammengefaßt werden, bleibt dem Benutzer überlassen. Im Unterschied zu

---

[4] Es ist sehr wohl möglich, Modelle der Mechanik zu erstellen, um bestimmte Aspekte der Software zu testen. Ein solcher Ansatz wird z.B. in [BT97] verfolgt. Als Grundlage für die Mechanik-Entwicklung reichen diese Möglichkeiten jedoch nicht aus.

[5] Ziel der Sprache EXPRESS-I ist die Definition von Testfällen für EXPRESS-Modelle, mit denen Konverter zwischen den Modellen und propietären Datenstrukturen auf ihre Konformität zum Standard überprüft werden können.

diesen Prinzipien führt die Strukturierung in Teilmodelle in der Sprache ROOM – wie oben diskutiert wurde – automatisch zu einer Zerlegung in (viele) Teilmodelle. Die Vererbungsmechanismen in UML und EXPRESS erlauben es darüber hinaus, wiederverwendbare Modulbibliotheken zu definieren bzw. zu nutzen. EXPRESS bietet hier umfangreichere Konzepte als die UML. Für die adäquate Modellierung der Transferstraße haben sich allerdings die Mechanismen der UML als ausreichend erwiesen. Die Verwendung der erweiterten Verknüpfungsoperatoren AND bzw. ANDOR in EXPRESS führt darüber hinaus in einigen Fällen darüber sogar zu schwer nachvollziehbaren Modellen, da die damit verbundenen Bedingungen durch Modul-Importbeziehungen aufgebrochen werden können.

Zusammenwirken der Teilsprachen

Die Vielzahl der zur Verfügung stehenden Beschreibungstechniken der UML, deren Anwendungsbereiche teilweise überlappen, führt vor allem zu einer Erhöhung des Einarbeitungsaufwandes. Besonders die Entscheidung, Zusammenspieldiagramme oder Sequenzdiagramme für die Modellierung der Abläufe auf Zellenebene einzusetzen, hat sich – wegen der beschriebenen wechselseitigen Vor- und Nachteile – als schwierig für die Bearbeiter erwiesen. Das Zusammenwirken der Beschreibungstechniken in ROOM hat sich für den Bearbeiter als unproblematisch erwiesen. Allerdings sind MSCs nur zum Teil in die Sprache integriert und konnten aus diesem Grund für die Modellierung von Interaktionen nicht verwendet werden, so daß das für die UML beschriebene Problem nicht auftreten konnte. Als problematisch wurde vom Bearbeiter die Einbeziehung von synchronen Kommunikationsmechanismen in die asynchrone Kommunikationsinfrastruktur der Sprache genannt. EXPRESS bietet lediglich eine einzige Notation für die Modellierung an, so daß für den Bearbeiter keine Schwierigkeiten durch die Kombination von Teilsprachen auftreten konnten. Die erweiterte Spezifikationssprache zur Darstellung von logischen bzw. algorithmischen Beziehungen wurde in der Fallstudie nicht verwendet, ebenso hat sich die Integration von EXPRESS-I aufgrund der genannten Nachteile nicht als sinnvoll erwiesen.

4.2 Bewertung der Werkzeugunterstützung

Im folgenden werden die Erfahrungen mit den in der Fallstudie eingesetzten Werkzeuge *Rational Rose*, *ObjecTime* und *EXPRESS-Modeler* zusammengefaßt. Für die UML und EXPRESS werden neben diesen Werkzeugen noch weitere am Markt angeboten, die aus Kapazitätsgründen jedoch nicht in die Fallstudie einbezogen werden konnten. Die gesammelten Erfahrungen können nicht generell auf Werkzeuge von anderen Herstellern übertragen werden. Gleichwohl wurde bei der Auswahl darauf Wert gelegt, jeweils am Markt führende Werkzeuge in der Studie zu testen.

Vollständigkeit des durch Werkzeuge unterstützten Sprachumfangs

Die Sprache ROOM ist unmittelbar mit dem Werkzeug *ObjecTime* des gleichnamigen Herstellers verbunden. Die Sprache wird in vollem Umfang von diesem Werkzeug unterstützt. Die im vorangegangenen Abschnitt diskutierte unvollständige Integration der MSCs entspricht der Definition der Sprache. Hilfreich bei der Erstellung der Modelle war insbesondere die Unterstützung der Simulation der entwickelten Modelle, die bei ROOM als einziger Sprache der Fallstudie möglich war.

Die Sprachen UML und EXPRESS werden dagegen nicht vollständig von den getesteten Werkzeugen unterstützt. Die Modellierung der Transferstraße mit der UML wurde jedoch von der eingeschränkten Werkzeugunterstützung nicht beeinträchtigt, da die betroffenen Sprachmittel – etwa die Teilsprache OCL – nicht benötigt wurden. Mit dem EXPRESS Modeler war es nicht möglich, die Kardinalität von Aggregationsbeziehungen zu spezifizieren sowie für Vererbungsbeziehungen die Verknüpfungsoperatoren AND und ANDOR zu spezifizieren. Die Verwendung der beiden Operatoren ist durchaus kritisch zu betrachten; ihre Anwendung wurde deshalb weitgehend vermieden. Dagegen stellt die fehlende Unterstützung der Kardinalität von Aggregationsbeziehungen eine schwerwiegende Einschränkung dar.

Eignung der Werkzeugunterstützung zur Dokumentation

Prinzipiell kann die graphische Darstellung der Modelle durch die benutzten Werkzeuge als Dokumentation benutzt werden. In der Praxis werden die erstellten Modelle jedoch häufig an mehreren Stellen des Entwicklungsprozesses benötigt. Die Verwendung der vollständigen Werkzeuge an allen betroffenen Arbeitsplätzen ist jedoch sowohl aus finanziellen Gründen als auch aufgrund der notwendigen Einarbeitung häufig nicht sinnvoll. Betrachtet werden aus diesem Grund hier die Funktionen zum Export der erstellten Modelle und deren Eignung für eine sinnvolle Weiterverwendung.

Der EXPRESS Modeler bietet eine Schnittstelle zum Export der Modelle im wmf-Format an, so daß die Modelle in Standard Textverarbeitungssysteme übernommen werden können. Allerdings sind bei der Übertragung der Modelle mehrfach Konvertierungsfehler aufgetreten, so daß die Schnittstelle sich für den Austausch großer Modelle als ungeeignet erwiesen hat.[6] Die Darstellung großer Modelle ist auch im Werkzeuge Rational Rose nicht völlig befriedigend gelöst. Angeboten wird lediglich eine Funktion für die Skalierung der Darstellungsgröße, die jedoch wichtige Modellinformationen wie Transitionsbedingungen in Zustandsübergangsdiagrammen bei großen Modellen ggf. nicht mehr darstellt. ObjecTime ermöglicht den Export der entstandenen Modelle im Postscript-Format. Damit sind die Modelle technisch unproblematisch weiterverwendbar. Nachteilig ist allerdings, daß auch hier wichtige Informationen wie Transitionsbedingungen generell nicht exportiert werden. Als Fazit muß festgestellt werden, daß derzeit keines der eingesetzten Werkzeuge eine ausreichende Unterstützung zur Dokumentation für die Weiterverwendung der Modelle außerhalb der Werkzeuge selbst bietet.

Benutzungsfreundlichkeit der Werkzeuge bei der Modellierung

Die Werkzeuge bieten einen sehr unterschiedlichen Funktionsumfang an. Betrachtet wurde in diesem Zusammenhang vor allem die Bedienerfreundlichkeit bei der Modellierung. Auf weitere Funktionalitäten, wie Simulationsfunktion des Werkzeugs ObjecTime, wird aufgrund der fehlenden Vergleichbarkeit nicht eingegangen. Der EXPRESS Modeler ermöglichte eine intuitive Bedienung. Kritisiert wurde hier lediglich die mitgelieferte Dokumentation, die bei der Klärung des unterstützten Sprachumfangs nicht weiterhelfen konnte. Positiv hervorzuheben ist dagegen die Unterstützung durch den Support des Herstellers. Das Werkzeug Rational Rose orientiert sich zwar bei der Bedienung an gängigen GUI-Standards. Trotzdem wurde vom Bearbeiter die unübersichtliche und z.T. umständliche Bedienung beklagt. Als Grund wurde hier vor allem genannt, daß aufgrund der großen Funktionalität häufig gebrauchte Funktionalitäten nur umständlich erreichbar sind. Das Werkzeug ObjecTime erfordert dagegen aufgrund der unüblichen Benutzeroberfläche einen erhöhten Einarbeitungsaufwand. Die Bedienung selbst erwies sich für den Bearbeiter nach der Einarbeitung als funktionell. Nachteilig war allerdings die bei umfangreichen Modellen große Anzahl an Teilmodellen, die sich in einer Vielzahl von Bedienungsfenstern niederschlägt und zur Unübersichtlichkeit führt.

Infrastruktur an Modellierungshilfen und Bibliotheken

Aufgrund des Anwendungshintergrunds der drei Sprachen existieren lediglich in EXPRESS in größerem Umfang Modelle aus dem Anwendungsfeld der Entwicklung mechatronischer Systeme. Die Teilnormen des STEP-Standards umfassen Modelle für die wesentlichen Informationen, die in der Konstruktion sowie in der Produktionstechnik benötigt werden. Für die Entwicklung mechatronischer Systeme von besonderer Bedeutung sind die Modellierung von Kinematikstrukturen (ISO 10303-105) sowie Geometriemodelle (ISO 10303-42). Ziel der Modelle ist die Standardisierung des Datenaustauschs. Die betreffenden Teilnormen sind aus diesem Grund in einer Reihe von bestehenden Engineering Werkzeugen, insbes. CAx-Systemen, integriert. In die konzeptionelle Modellierung von Anlagen konnten die Teilnormen jedoch sowohl aufgrund der Komplexität als auch aufgrund der

[6] Der Student, der die Fallstudie bearbeitet hat, ist aus diesem Grund sogar auf ein Desktop Publishing Programm zur Darstellung dieser Modelle ausgewichen.

mangelnden Eignung für die benötigten Anwendungen nur unbefriedigend eingebunden werden.[7] Die Sprachen UML und ROOM werden in der Anlagenmodellierung bislang nur wenig bzw. nur in spezifischen Anwendungen eingesetzt. Auf existierende Modelle konnte damit in der Fallstudie nicht zurückgegriffen werden. Insbesondere in der UML konnten jedoch gut eigene Bibliotheken für Kinematikstrukturen aufgebaut werden.

4.3 Potential für Integration der drei Sprachen

Die in den vorangegangenen Abschnitten besprochenen Ergebnisse der Fallstudie zeigen, daß keine der untersuchten Sprachen derzeit eine ausreichende Unterstützung bietet, um alle Aspekte der Anlagenplanung abdecken zu können. Gleichwohl kann eine Vielzahl der zu modellierenden Informationen mit mindestens einer der untersuchten Sprachen adäquat dargestellt werden. Aus diesem Grund wird im folgenden kurz Ansatzpunkte für eine mögliche Integration der Sprachen diskutiert.

Offensichtlich ist die Ähnlichkeit von EXPRESS mit UML-Klassendiagrammen. Beide Sprachen stellen Aggregations- und Vererbungsbeziehunen in den Mittelpunkt. Für die grundlegenden Konzepte beider Sprachen wird in [AP98] gezeigt, daß hier eine einfache Abbildung möglich ist. Allerdings definieren beide Sprachen eine Reihe von spezialisierten Modellierungskonstrukten, die keine Entsprechung in der jeweils anderen Sprache haben und deren Abbildung einer eingehenden Untersuchung bedarf. Beispiele sind Verknüpfungsoperatoren für Vererbungsbeziehungen sowie feingranulare Modulimport-Beziehungen in EXPRESS sowie Stereotypes in der UML. Konzepte für eine vollständige Abbildung zwischen beiden Sprachen werden derzeit in der Arbeit [Ler00] untersucht. Eine Integration von EXPRESS und UML-Klassendiagrammen bietet einerseits den Vorteil, die Werkzeug-Infrastruktur des Maschinenbaus unmittelbar nutzen zu können und andererseits die in der Software Entwicklung wesentlichen Ablaufaspekte modellieren zu können. Weniger direkt ist eine Einbindung von ROOM in eine integrierte Modellierung möglich. ROOM Strukturdiagramme beschreiben konkrete Objektbeziehungen, während in EXPRESS sowie in Klassendiagrammen Beziehungen zwischen Klassen von Objekten beschrieben werden. Die Integration der beiden Konzepte ist Gegenstand von Arbeiten, auf Basis von ROOM eine Echtzeit Variante der UML zu entwickeln [SR98]. Der Ansatz, der in diesem Zusammenhang verfolgt wird, modelliert ROOM Aktordiagramme mittels Zusammenspieldiagrammen mit Sprachmitteln der UML. Die methodische Integration beider Sprachen ist jedoch bislang kaum untersucht worden.

5 Zusammenfassung und Diskussion der Ergebnisse

In einer vergleichenden Fallstudie werden maßgebliche Modellierungssprachen aus den Anwendungsbereichen Maschinenbau und Software Entwicklung hinsichtlich ihrer Eignung in einer integrierten Entwicklung von mechatronischen Systemen untersucht. Mit EXPRESS, UML und ROOM wurden führende Sprachen aus beiden Anwendungsbereichen in die Fallstudie einbezogen. Die Ergebnisse der Fallstudie zeigen, daß keine der betrachteten Sprachen alle wesentlichen Aspekte einer integrierten Systemmodellierung adäquat darstellen kann. Der Vergleich zeigt aber auch, daß sich die drei Sprachen aufgrund ihres unterschiedlichen Anwendungshintergrunds gut ergänzen.

Hinsichtlich der Ausdrucksmächtigkeit der Sprachen zur Modellierung von Systemstrukturen haben sich keine wesentlichen Unterschiede ergeben. Alle Sprachen können allgemeine topologische Beziehungen wie Informations- und Materialfluß visuell darstellen, nicht jedoch Geometrie- und Kinematikstrukturen. EXPRESS bietet hier den Vorteil, daß bei Benutzung der entsprechenden STEP-Normen die Anbindung an Engineering Werkzeuge des Maschinenbaus, insb. CAx- Systeme, möglich ist. Die Aktor-Diagramme in ROOM bieten dagegen die beste Darstellung von Material- und Informationsflußbeziehungen im realen System. Zur Darstellung von Systemabläufen ist EXPRESS

[7] Der Anwendungshinergrund der Modelle, einen Standard für den technischen Datenaustausch zur Verfügung zu stellen, führte teilweise zu Problemen beim Einsatz in der konzeptionellen Modellierung.

bekanntermaßen nicht konzipiert. Der Abstraktionsgrad der Darstellungsmittel von Systemabläufen, die in UML und ROOM angeboten werden, ist vor allem für die Planungsphasen der Entwicklung mechatronischer Systeme geeignet. Die UML ermöglicht als einzige Sprache die vollständige Modellierung von Interaktionsszenarien und bietet mit einer Reihe von Teilsprachen eine große Flexibilität. Dieser Vorteil wird jedoch durch den hohen Einarbeitungsaufwand und die teilweise Überlappung der Anwendungsbereiche, bei der viele Fragestellungen der Integration und Konsistenz noch ungelöst sind, wieder relativiert. Die Darstellungsmittel von ROOM sind vorwiegend für den konstruktiven Systementwurf, insbesondere des Softwareanteils, gut geeignet. Die Strukturierung der ROOM-Modelle orientiert sich am besten an der Zerlegung des realen Systems, während die Verwendung von EXPRESS bzw. von UML-Klassendiagrammen zu einer Vermengung von konzeptionellen und konkreten Strukturen führt.

Die angebotene Werkzeugunterstützung für die Sprachen setzt entsprechend des vorwiegenden Einsatzbereiches sehr unterschiedliche Akzente. Während der EXPRESS-Modeler ein reines Modellierungswerkzeug darstellt, bieten das ROOM Werkzeug ObjecTime und das UML Werkzeug Rational Rose auch Möglichkeiten zur Simulation von Modellen bzw. zur Code Generierung[8]. Anlaß zur Kritik gaben allerdings alle in der Fallstudie betrachteten Werkzeugebeim Export der entwickelten Modelle als Dokumentation zur Weiterverarbeitung in anderen Werkzeugen. In allen Werkzeugen konnten große Modelle nicht adäquat bzw. fehlerfrei exportiert werden. Die Kritikpunkte reichen von der fehlenden Darstellung von Transitionsbedingungen in Zustandsübergangsdiagrammen bis zu gravierenden graphischen Fehlern. Alle Werkzeuge erfordern einen erheblichen Einarbeitungsaufwand, der zum Teil von der umfangreichen Funktionalität, aber auch durch zu umständliche bzw. ungewohnte Bedienerführung verursacht wird. Eine Unterstützung des Modellierers durch wiederverwendbare Modellbausteine existiert bislang nur in EXPRESS in Form der STEP-Normen. Allerdings orientieren sich diese Modelle primär an den Erfordernissen des automatisierten Datenaustauschs zwischen Engineering Werkzeugen und können häufig nicht direkt beim Entwurf von mechatronischen Systemen verwendet werden.

Ein wesentliches Ergebnis der Fallstudie ist, daß keine der betrachteten Modellierungssprachen, die als stellvertretend für den Stand der Technik in den Bereichen Software Entwicklung und mechanische Konstruktion gelten können, alleine ausreicht, um eine integrierte Modellierung von mechatronischen Systemen zu ermöglichen. Da sich die Sprachen in ihrer Ausdrucksmächtigkeit jedoch gut ergänzen, erscheint gerade für die Planungsphasen im Entwicklungsprozeß eine Integration der Sprachen gewinnbringend. Aufgrund der konzeptionellen Ähnlichkeit der Sprachen erscheint insbesondere eine Integration von EXPRESS und der UML auf Basis einer Übersetzung zwischen EXPRESS-G und UML Klassendiagrammen sinnvoll. Auch eine Einbindung der Konzepte von ROOM erscheint aufgrund der guten Eignung zur Darstellung von Systemstrukturen hilfreich. Ansätze hierfür, die allerdings noch wichtige methodische Fragestellungen offen lassen, bietet die Integration von ROOM- und UML-Konzepten in der UML-RT. Eine integrierte Modellierung mechatronischer Systeme in der Planungsphase erscheint daher auf Basis existierender Beschreibungstechniken realisierbar, erfordert jedoch die Klärung einer Reihe von technischen und methodischen Fragestellungen.

Danksagung: Im Rahmen der Durchführung und Auswertung der Fallstudie haben wir viele hilfreiche Diskussionen mit unserem Kollegen Gunnar Billing, FAU Erlangen geführt. Zu Dank verpflichtet sind wir ebenfalls Alexander Sabbah (iwb) und natürlich den Studenten, die die Fallstudie bearbeitet haben: Marco Hoffmann, Martin Kacalek und Bernd Lercher. Ohne ihren Einsatz hätte diese Arbeit nicht entstehen können.

[8] Die Codegenierierung von Rational Rose umfaßt lediglich die automatische Erzeugung von Programmstrukturen, während das Werkzeug ObjecTime ablauffähige Softwaresysteme generiert. Von Drittherstellern werden jedoch auch für Rational Rose Erweiterungen angeboten.

6 Literatur

[AP98] Arnold F., Podehl G., Best of Both Worlds – A Mapping from Express G to UML, UML'98, 1998

[BM99] Billing G., Mauderer M. Integrated Planning and Development of Manufacturing Systems. In 15th International Conference on Production Research (ICPR). 1999.

[Boo94] G. Booch94. Object Oriented Analysis and Design with Applications. The Benjamin/Cummings Publication Company, 1994.

[BT97] K. Bender, J. Tomaszunas. Erfahrungsbericht zur qualitativen Modellierung mechanischer Systeme zum Steuerungstest in Echtzeit. In: EKA'97, Automatisierungssysteme (Kongreßband), Braunschweig, 1997.

[Chen76] P. Chen. The Entity-Relationship Model - Towards a Unified View of Data. In: ACM Transactions on Database Systems, Vol. 1, No. 1, pp 9-36, 1976.

[Dou97] Douglass B. P. Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks an Patterns. Addison Wesley, 1997.

[Har87] Harel D. Statecharts: A Visual Formalism for Complex Systems. In Science of Computer Programming 8, pg. 231. 1987.

[Hof98] Hoffmann M. "Vergleichende Fallstudien zur gesamtheitlichen Modellierung von Werkzeugmaschinen STEP/EXPRESS", interdisziplinäres Projekt, FORSOFT A4/D1, 1998.

[ISO94a] International Standards Organisation (ISO). Product data representation and exchange, Part 1: Overview and fundamental principles. ISO-Norm, Beuth Verlag, 1994.

[ISO94b] International Standards Organisation (ISO). Product data representation and exchange, Part 11: Description methods: The EXPRESS language reference manual. ISO-Norm, Beuth Verlag, 1994.

[ISO94c] International Standards Organisation (ISO). Product data representation and exchange, Part 11: Description methods: The EXPRESS-I language reference manual. ISO-Norm, Beuth Verlag, 1994.

[ISO94d] International Standards Organisation (ISO). Product data representation and exchange, Part 42: Integrated generic resources: Geometric and topological representation. ISO-Norm, Beuth Verlag, 1994.

[ISO94e] International Standards Organisation (ISO). Product data representation and exchange, Part 105: Integrated application resource: Kinematics. ISO-Norm, Beuth Verlag, 1994.

[ISO94f] International Standards Organisation (ISO). Product data representation and exchange, Part 203: Application Protocol: Configuration controlled design. ISO-Norm, Beuth Verlag, 1994.

[ITU93] ITU-T. Recommendation Z.100. Specification and Description Language (SDL). ITU, 1993.

[ITU96] ITU-TS. Recommendation Z.120: Message Sequence Chart (MSC). Geneva, 1996.

[Jac92] I. Jacobson. Object Oriented Software Engineering: A Use Case Driven

Approach. Addison Wesley, 1992.

[Kac98] Kacalek M. "Vergleichende Fallstudien zur gesamtheitlichen Modellierung von Werkzeugmaschinen UML", interdisziplinäres Projekt, FORSOFT A4/D1, 1998.

[KvK+99] Kamsties E., v. Knethen A., Philipps J., Schätz B. Eine vergleichende Fallstudie mit CASE-Werkzeugen für formale und semi-formale Beschreibungstechniken. In: Formale Beschreibungstechniken für verteilte Systeme. 9. GI/ITG Fachgespräch. Herbert Utz Verlag. 1999.

[Ler98] Lercher B. "Vergleichende Fallstudie zur gesamtheitlichen Modellierung von Werkzeugmaschinen ROOM", interdisziplinäres Projekt, FORSOFT A4/D1, 1998.

[Ler00] Lercher B. "Konzeption einer Integration von STEP/EXPRESS und UML Modellen – Modellierung eines Werkzeugwechslers", Diplomarbeit, FORSOFT A4/D1, geplant 2/00.

[Lyo98] Lyons A. UML for Real Time Overview. http://www.objectime.on.ca/otl/technical/umlrt.html. April 1998.

[PB93] G. Pahl, W. Beitz. Konstruktionslehre - Methoden und Anwendung. Lehrbuch, 3. Auflage, Springer Verlag, Berlin, 1993.

[Rod91] W. G. Rodenacker. Methodisches Konstruieren. Lehrbuch, 4. Auflage, Springer Verlag, Berlni, 1991.

[RBP+91] J. Rumbaugh91, M. Blaha, W. Premerlani, F. Eddy und W. Lorensen. Object Oriented Modeling and Design. Prentice Hall, 1991.

[SGW94] B. Selic, G. Gullekson, P. Ward. Real-Time Object-Oriented Modeling. Wiley & Sons, New York, 1994.

[SR998] Selic B., Rumbaugh J. Using UML for Modeling Complex Real-Time Systems. http://www.objectime.on.ca/otl/technical/umlrt.pdf. März 1998.

[UML97a] Rational. UML Summary, Version 1.1. http://www.rational.com/uml/resources/documentation. September 1997.

[UML97b] Rational. UML Notation Guide, Version 1.1. http://www.rational.com/uml/resources/documentation. September 1997.

[UML97c] Rational. UML Semantics, Version 1.1. http://www.rational.com/uml/resources/documentation. September 1997.

[VDI77] VDI. VDI-Richtlinie 3260: Funktionsdiagramme von Arbeitsmaschinen und Fertigungsanlagen. 1977.

[WM85] Ward P.T., Mellor S.J. Structured Develompent for Real-Time Systems. Volume 1: Introduction and Tools. Yourdon Press, 1985.