Technische Universität Dresden
Bereich Ingenieurwissenschaften
Fakultät Informatik
Institut Für Angewandte Informatik
Professur Prozesskommunikation

Dissertationsschrift

Integriertes System- und Dienste-Management in der industriellen Automation

Vorgelegt von

Dipl.-Inf. Robert Lehmann

Zum Erlangen des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

Gutachter 1. Univ.-Prof. Dr.-Ing. habil. Martin Wollschlaeger

2. Ao. Univ. Prof. Dr. Wolfgang Kastner

Datum der Einreichung: 05.04.2016

Datum der Verteidigung: 12.10.2016

Vorwort

Alle wesentlichen Inhalte dieser Arbeit sind in meiner Zeit als wissenschaftlicher Mitarbeiter an der Professur Prozesskommunikation der Technischen Universität Dresden entstanden. Entsprechend dankbar bin ich für die eingeräumten Freiheiten in der Wahl und Ausprägung des Themas, das letztlich in dieser Arbeit niedergeschrieben ist.

Besonderer Dank gebührt Prof. Dr.-Ing. habil. Martin Wollschlaeger, der auch in schwierigen und unruhigen Zeiten stets Zeit fand und mit wertvollen Ratschlägen zur Seite stand.

Ich danke Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Wolfgang Kastner, Technische Universität Wien, für seine Arbeit als Gutachter dieser Arbeit.

Meinen Kollegen an der TU Dresden, allen voran Herrn Roman Frenzel, möchte ich für die fortwährenden Diskussionen über Jahre hinweg danken. Ohne sie wäre vieles im Unklaren geblieben. Auch den Studenten die im Rahmen ihrer Arbeiten Detailthemen betrachtet haben möchte ich danken.

Nicht zuletzt gilt mein ganz persönlicher Dank meiner Familie. Ohne ihre Unterstützung über Jahre hinweg hätte diese Arbeit nicht entstehen können.

Zusammenfassung

Die Industrie ist im Wandel. Die Grenzen zwischen Industrien, Anwendungsbereichen und Unternehmen verschwinden immer weiter, sind teils kaum noch in ihrer alten Ausprägung zu erkennen. Auch die industrielle Automation kann und sollte sich diesem Trend nicht entziehen. Immer mehr Technologien und Paradigmen anderer Bereiche gewinnen an Bedeutung. Hinzu kommt, dass die Anzahl und die Vielfalt an Geräten, Anwendungen, Anforderungen und Technologien stetig wächst, Fakten die in unzähligen Veröffentlichungen hervorgehoben werden.

Diese Arbeit befasst sich mit Ansätzen, die es ermöglichen, einigen Aspekten der wachsenden Komplexität zu begegnen. Dabei handelt es sich um Technologien und Konzepte zum Thema Management, genauer zum Netzwerk-, System- und Dienste-Management. Ziel ist es nicht nur einen Ansatz zu finden, der gegenwärtigen Ansprüchen genügt, sondern auch noch für kommende Entwicklungen geeignet ist. Was es bedeutet gegenwärtigen und zukünftigen Ansprüchen zu genügen, wird anhand von noch zu definierenden Kriterien bewertet.

Das Netzwerk-, System- und Dienste-Management ist, je nachdem auf welchen Bereich der Industrie man seinen Fokus richtet, sehr unterschiedlich ausgeprägt. Während in der klassischen Unternehmens-IT oder auch in der Telekommunikationsindustrie eine Vielzahl von spezifischen und etablierten Technologien zum Einsatz kommen, so sind in der industriellen Automation bislang wenig Ansätze etabliert, die den wachsenden Ansprüchen genügen können. Es stellt sich also die Frage: Können die Technologien zum Netzwerk-, System- und Dienste-Management aus anderen Industriebereichen für die industrielle Automation einfach übernommen werden, sind Adaptionen notwendig oder ist gar eine vollständige Neuentwicklung erforderlich? Um diese Frage zu beantworten, werden in der Arbeit Management-Ansätze aus Unternehmens-IT, Telekommunikation und industrieller Automation analysiert und gegenübergestellt. Insbesondere werden auch Vor- und Nachteile von Neuentwicklung und Adaption bestehender Technologien diskutiert.

Folglich wird, auch anhand nichttechnischer Kriterien, der entsprechende Entwicklungspfad ausgewählt und gemäß der Bedürfnisse der industriellen Automation ausgeprägt. Im Vordergrund stehen dabei die Ent- bzw. Weiterentwicklung von Informationsmodellen zum System-Management. Der Nachweis der Leistungsfähigkeit des gewählten Ansatzes erfolgt anhand von Anwendungsfällen, die praxisnahe Probleme im System-Management der industriellen Automation darstellen.

Im Verlauf der Arbeit werden immer wieder Bezüge zu angrenzenden Themen hergestellt und diskutiert. So etwa Notwendigkeiten von Standardisierung, aktuelle Entwicklungen in Bezug zu Industrie 4.0 und Wechselwirkungen zu den eigentlichen Mehrwert erzeugenden Prozessen der industriellen Automation. Abschließend werden Potentiale für zukünftige Entwicklungen diskutiert, die während der Arbeit offen bleiben mussten.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

System-Managment verfolgt heute nach wie vor das Ziel die Administration in immer komplexer werdenden Systemen zu ermöglichen oder zu erleichtern. Einst entstanden, um die Ansprüche großer Telekommunikationsunternehmen und ihrer Kunden an einen reibungslosen, sicheren und nachvollziehbaren Betrieb zu erfüllen, sind Aspekte des System-Managements heute ubiquitär.

Auch die Komplexität vernetzter Installationen in der industriellen Automation ist einem stätigen Wachstum unterworfen. Dies gilt für die Anzahl an verbauten und betriebenen Geräten genauso wie für jedes Gerät im Einzelnen. Es ist bei aktuellen Entwicklungen zu Industrie 4.0 und Cyber Physical Systems davon auszugehen, dass sich dieser Trend noch verstärken wird.

Das System-Management in der industriellen Automation verfolgt bislang jedoch vorrangig vollkommen andere Ansätze als es beispielsweise in der IT der Fall ist. Im Mittelpunkt stehen das hersteller- bzw. das technologie- oder protokollzentrierte Management. Ein übergreifendes und durchgängiges Management findet nur in wenigen Fällen statt und beschränkt sich dann im Wesentlichen auf die „IT-Aspekte" der industriellen Automation.

Es ist zurzeit nahezu ungeklärt, wie und auf Basis welcher Technologien und Paradigmen das System-Management in der Industrie in der Zukunft erfolgen kann.

1.1 Definition der Zieldomäne

Der Betrachtungsgegenstand dieser Arbeit ist der technologische Bereich, in dem Prozesse oder Einrichtungen unter festgelegten Bedingungen ohne menschliches Eingreifen, also automatisch, ablaufen oder arbeiten [1]. Dieser Bereich wird allgemein als Automation bezeichnet.

Automation umfasst eine ganze Reihe von Gebieten, darunter die Gebäudeautomation, die Verkehrsautomation und die industrielle Automation. Alle Betrachtungen, die im Rahmen dieser Arbeit durchgeführt werden, befassen sich ausschließlich mit der industriellen Automation. Die Begriffe (industrielle) Automation oder Automatisierung werden synonym für industrielle Automation verwendet. Eine weitere Aufteilung, etwa in Prozess- und Fabrikautomation, wird nicht vorgenommen. Die technologischen Ansätze der beiden Bereiche unterscheiden sich nicht in einem Umfang, in dem grundsätzliche Prinzipien des System- und Dienste-Managements anders angewendet werden müssten.

Anlagen, beziehungsweise Systeme, die gegenwärtig zum Zweck der Automatisierung von Abläufen betrieben werden, unterscheiden sich teils stark. Die Methoden, die in dieser Arbeit entwickelt und angewendet werden, eignen sich vor allem für moderne Anlagen. Damit sind all die Installationen gemeint, die einen hohen Grad an Kommunikation und Vernetzung zwischen den einzelnen automatisierenden Komponenten aufweisen. Detaillierte Betrachtungen zur Automation und ihren Eigenheiten sowie technologischen Trends werden in Abschnitt 1.3 durchgeführt.

1.2 Ziel der Arbeit

Die Möglichkeiten, Herausforderungen und Ansprüche eines einheitlichen System- und Dienste-Managements für die industrielle Automation zu evaluieren ist das erste der beiden wissenschaftlichen Kernziele der Arbeit. Dafür müssen im Detail bestehende Ansätze zum System-Management, egal aus welchem Bereich sie stammen, hinsichtlich ihrer Eignung analysiert und bewertet werden, um so eine Aussage bezüglich genereller Eignung, Zukunftssicherheit, Erweiterungsbedarf, Reifegrad und Verfügbarkeit treffen zu können. Damit eine systematische Bewertung umgesetzt werden kann, sind Kriterien auf Basis von Anforderungen zu definieren.

Das zweite wissenschaftliche Kernziel ist die Entwicklung eines System- und Dienste-Managements für die industrielle Automation. Dabei wird es von den Aussagen zur Eignung bestehender Ansätze abhängen, ob eine Neu- oder Weiterentwicklung stattfinden. Unabhängig davon wird die entsprechende Entwicklung von Informationsmodellen zum System- und Dienste-Management in der industriellen Automation im zweiten Teil im Vordergrund stehen.

Der Nachweis der praktischen Umsetzbar- und Leistungsfähigkeit des zu wählenden Ansatzes stellt den letzten wissenschaftlichen Beitrag der Arbeit dar. Hierfür werden in einer konkreten Umgebung einzelne Anwendungsfälle, deren Relevanz und Repräsentativität zu diskutieren ist, explizit umgesetzt.

1.3 Industrielle Automation

Das Automatisieren von Abläufen ist keine Entwicklung der letzten ein- oder zweihundert Jahre, sondern vielmehr ein Prozess, der fortwährt: angefangen von einfachen, von Menschen bewusst oder unbewusst ausgeführten Automatismen, über die Erfindung des Buchdrucks, die industrielle Revolution im 19. Jahrhundert bis hin zu modernen industriellen Anlagen, die immer weiter das Ziel einer vollständigen Automation verfolgen. Diese modernen Anlagen können mehrere tausend Geräte umfassen, die miteinander kommunizieren – und das in Anlagen, die sich über Quadratkilometer erstrecken oder Fabrikhallen füllen und Anforderungen an die Kommunikation im Millisekundenbereich stellen.

Die industrielle Automation hat sich mehrere Jahrzehnte lang, in vielerlei Hinsicht unbeeindruckt von Entwicklungen im Telekommunikations-, Unternehmens- oder Bürokommunikations-Sektor, eigenständig entwickelt. Technologische sowie organisatorische Eigenheiten, die sich im Laufe der Zeit dabei herausgebildet haben, werden in diesem Abschnitt

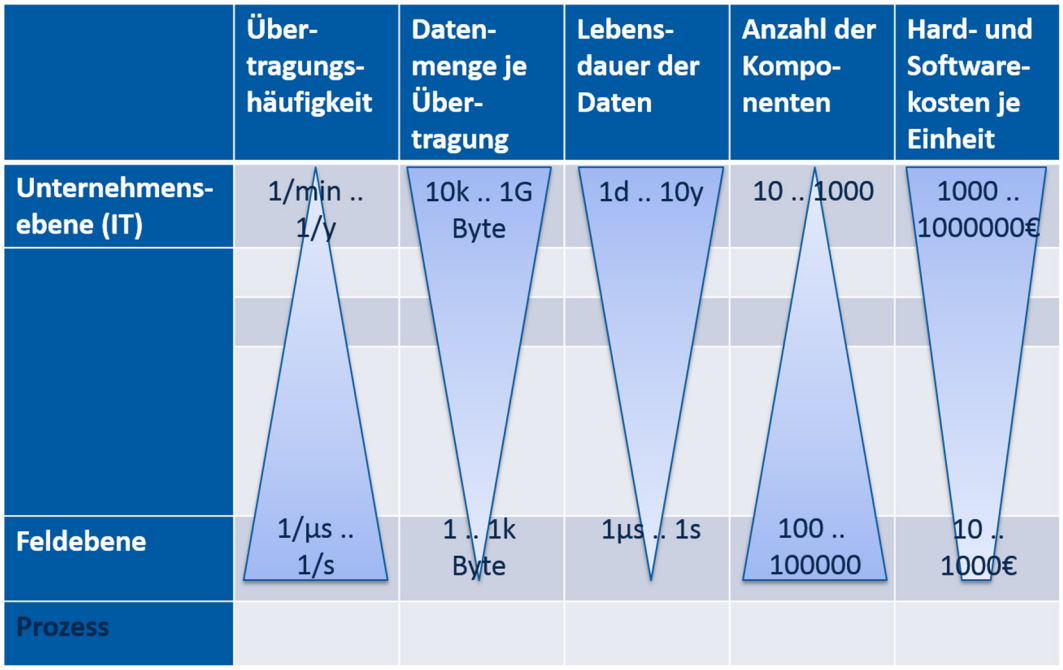

Abbildung 1 Eigenschaften industrieller Kommunikationssysteme

betrachtet und sich abzeichnende oder schon gelebte Trends werden vorgestellt. Abbildung 1 stellt einen Teil der Eigenheiten der industriellen

Automation dar und versucht ungefähre Größenordnungen zu vermitteln. Je nach betrachteter Technologie ändern sich die dargestellten Werte, die Verhältnisse zwischen den Ebenen bleiben jedoch bestehen.

1.3.1 Organisatorische und technologische Eigenheiten der industriellen Automation

Organisatorische Eigenheiten

Eine Vielzahl von Herstellern auf der einen und Integratoren und Betreibern auf der anderen Seite prägen die Organisationsstrukturen in der industriellen Automation. Im Allgemeinen organisieren sich sowohl Hersteller wie auch Integratoren und Endanwender in Nutzerorganisationen (auch Feldbusorganisationen, Vendor-Associations). In der Regel ist es so, dass, vor allem Hersteller zur Abdeckung von Markterfordernissen, Mitglied in mehr als einer Organisation sind und auch Produkte im Rahmen der Vorgaben mehrerer Organisationen fertigen. Fast jede dieser Organisationen definiert eigene Protokolle für die Kommunikation zwischen industriellen Geräten. Dabei handelt es sich nicht nur um Protokolle der Anwendungsebene sondern auch um grundlegende Kommunikations- und Übertragungsprotokolle sowie vollkommen unterschiedliche Übertragungsphysiken. Die Kombinationen vieler Hersteller mit einer großen Zahl an industriellen Kommunikationsprotokollen ergibt das stark heterogene Bild, in dem sich die industrielle Automation heute darstellt. Ein Eindruck über die Organisationsstrukturen, Zusammenhänge und Abhängigkeiten zwischen den einzelnen Feldbusorganisationen und Protokollen wird in [2] gegeben.

Eine der wesentlichen organisatorischen Eigenheiten der Automation ist die nahezu vollständige Planung und Konfiguration (Anlagen-Engineering) einer (Teil-)Anlage bevor diese real existiert. Natürlich wird auch in der Enterprise-IT, vor allem für umfangreiche Installationen, im Vorfeld geplant. Das in der Automation verfolgte Top-Down-Konzept [3] ist sehr viel umfassender, vor allem aber nahezu unabhängig von der erwarteten Anlagengröße. Selbst verhältnismäßig kleine Installationen

werden vollständig offline geplant und auch programmiert. Durch das vollständige Offline-Engineering von Anlagen hat sich innerhalb der Automation die Wahrnehmung gefestigt, dass die Planungsdaten zu jedem Zeitpunkt den realen Zustand der Anlage wiederspiegeln, auch nach Jahren des Betriebs. In der Praxis erweist sich diese Annahme jedoch oft als nicht haltbar.

Auf Ebene der logischen Strukturen zur Kommunikation werden automatisierungstechnische Anlagen konsequent segmentiert. Komponenten, die auf die Kommunikation untereinander angewiesen sind, müssen sich entsprechend im gleichen Netzwerksegment befinden. Eine vollständige Entkopplung der Segmente ist darunter nicht zu verstehen, selbstverständlich gibt es Kommunikationskanäle zwischen diesen Segmenten (Zellen), beispielsweise für Konfiguration, Administration und generell für die Kommunikation mit den höheren Schichten der Automatisierungs-Pyramide (Abbildung 2). Die wesentliche Nutzdatenkommunikation findet aber innerhalb eines Segmentes und nur in Ausnahmefällen über die jeweiligen Zellengrenzen hinweg statt. In der Regel wird kein Routing im klassischen Sinn durchgeführt (siehe technologische Eigenheiten).

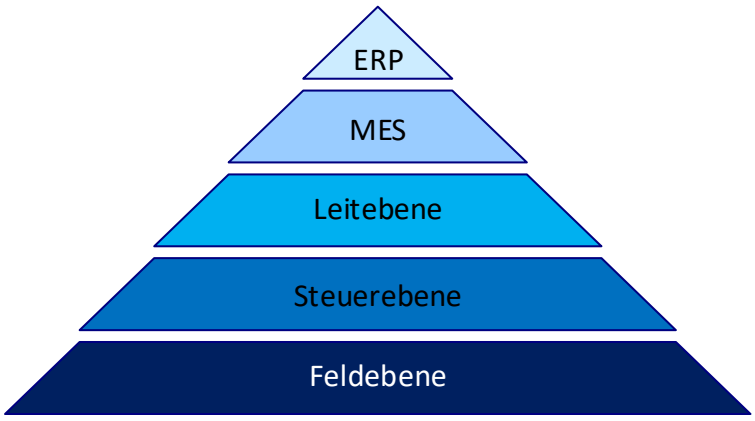

Abbildung 2 Automatisierungspyramide

Die Trennung der industriellen Automation in Prozess- und Fertigungsautomation (PA und FA) ist in der Wahrnehmung der Experten etabliert. Grundlegend unterscheiden sich diese beiden Bereiche vor allem in Anlagen-Laufzeiten (FA geringer als PA), Einsatzbedingungen (FA

Fabrikhalle, PA innerhalb chemischer oder verfahrenstechnischer Prozesse) und Ablaufgeschwindigkeit (FA hohe Zeitanforderungen, PA eher träge). Mit den Anforderungen, die durch das jeweilige Einsatzgebiet gestellt werden, gehen Anforderungen gegenüber den eingesetzten Geräten und Technologien einher. Die teilweise großen Schwankungen der in Abbildung 1 auf der Feldebene dargestellten Werte sind Ergebnis der abweichenden Anforderungen zwischen PA und FA.

Eine letzte wesentliche Eigenschaft der industriellen Automation bzw. einiger Branchen wie Prozessindustrie oder Energieerzeugung sind die erheblichen Laufzeiten von Anlagen und damit verbunden auch die Laufzeiten der verbauten Gerätschaften. Solche Anlagen und Geräte können leicht Laufzeiten von mehreren Jahrzehnten erreichen. Es ist gelebte Praxis, dass die verbauten Geräte erst bei Außerbetriebsetzung oder Defekt das nächste Mal in die Hand genommen werden. Modernisierungen (bezogen auf Einzelkomponenten oder Soft-/Firmware) finden in aller Regel nicht statt und sind aus Sicht der Automatisierer in vielerlei Hinsicht auch nicht notwendig bzw. aus regulatorischer Sicht oder von Rechtswegen nicht ohne weiteres möglich.

Technologische Eigenheiten

Will man die technologischen Eigenheiten der industriellen Automation mit wenigen Worten beschreiben, kann man sagen: Die industrielle Automation ist geprägt von Standardisierung und Heterogenität. Im ersten Moment scheinen beide Begriffe nicht zusammen zu passen, denn Standardisierung soll grundsätzlich eine allzu große Heterogenität innerhalb eines Betrachtungsraumes verhindern. In der Automation sind viele Technologien - und letztlich die sie definierenden Standards – jedoch unabhängig voneinander in einzelnen Branchen und vor allem auch Regionen der Welt gewachsen. So existieren zum Beispiel gegenwärtig mehrere industrielle Kommunikationsprotokolle, die auf die Steuerung von höchst zeitkritischen Antrieben ausgelegt sind und diese Aufgabe, nach Einschätzung des Marktes, auch lösen können. Nahezu alle diese Protokolle werden noch gepflegt. Über die Jahre haben sich so für sehr ähnliche Aufgaben verschiedene Protokolle etabliert, die technologisch aber keineswegs ohne zusätzliche technische Aufwände

kompatibel zueinander sind. Dabei beschränkt sich die Inkompatibilität nicht nur auf die Kommunikationsprotokolle, sondern bezieht sich auch auf Anschlüsse (etwa Steckverbindungen), Elektrik und Zugriffs- sowie Organisationskonzepte. Wie erwähnt, organisieren Interessenverbände die Weiterentwicklung und das Fortbestehen der verschiedenen industriellen Kommunikationsprotokolle. An der Heterogenität können die Interessenverbände jedoch nur bedingt etwas ändern, das Gegenteil ist in der Realität eher der Fall.

Die Datenkommunikation in der industriellen Automation trennt zwischen zyklischer und azyklischer Übertragung. Bei der zyklischen Kommunikation handelt es sich um die eigentliche Nutzdatenkommunikation im Sinne einer automatisierungstechnischen Anwendung. Die zyklische Kommunikation ist geprägt von einer großen Anzahl an vergleichsweise kleinen Datenpaketen (Abbildung 1), die mit einem im Vorfeld eingestellten Zeitverhalten zwischen einzelnen Geräten ausgetauscht werden. Alle anderen Informationen werden azyklisch (anfrage- und ereignisbasiert) übertragen, das schließt neben Konfiguration und Administration auch die Kommunikation zwischen Feldebene und Unternehmensebene mit ein. Bedingt durch die erwähnten üblichen Strukturen (Segmentierung) werden wesentliche Teile des Nutzdatenverkehrs, auch in Ethernet basierten Automatisierungsnetzen, nicht geroutet.

Auf Ebene der im Feld eingesetzten Geräte ist die Automation geprägt von einer großen Anzahl (pro Installation) an verschiedensten Geräten. Bis auf Ausnahmen (Industrie PCs) handelt es sich dabei um Embedded-Systeme, die in der Automation besonders von geringen Kapazitäten (Speicher, Ausführungskapazitäten etc.) geprägt sind.

1.3.2 Technologische Trends in der industriellen Automation

Die gegenwärtige Entwicklung der Automation ist in vielen Bereichen immer noch stark durch ihre Wurzeln geprägt. Aufgrund der historisch gewachsenen heterogenen Technologielandschaft auf der einen und den sich ändernden Anforderungen auf der anderen Seite sind seit einigen

Jahren große Integrationsbemühungen auf verschiedenen Ebenen zu erkennen. Auf der Feldebene wird intensiv versucht, die vielen verschiedenen Kommunikationsansätze zu integrieren. Zwischen den Ebenen der Automatisierungspyramide und sogar übergreifend über Firmengrenzen hinweg werden ebenfalls verschiedene Bemühungen unternommen, um einen ungehinderten Austausch von Daten zu erreichen.

Der wesentliche Integrationstrend ist aber der zwischen Automation und IT allgemein. Die vordergründig ist dabei ist die verstärkte Verwendung von Technologien und Ansätzen, die im Wesentlichen direkt aus der Enterprise IT stammen. Ethernet und die zugehörigen Protokolle und Strukturen sind die treibenden Kräfte hinter dem aktuellen Trend des Zusammenwachsens der Automation und IT. Zwar wurden IT-Technologien, darunter auch Ethernet, schon lange im Kontext der Automation eingesetzt, nicht jedoch direkt für die Kommunikation zwischen Feldgeräten. Dieser Bereich der Kommunikation war jahrzehntelang den klassischen Feldbussen vorbehalten. Die Ethernet-basierte Kommunikation zwischen Feldgeräten wird häufig als „*Industrial Ethernet*" bezeichnet, um eine Abgrenzung gegenüber dem *normalen* Ethernet auszudrücken. Der mittlerweile weitreichende Einsatz von Ethernet in der Automation hat auch dazu geführt, dass weitere Internettechnologien (z.B. XML, SNMP, Web Services etc.) ihren Weg in die Industrie gefunden haben und sich nun einer wachsenden Bedeutung und Beliebtheit erfreuen. Der Grad der durchgängigen Vernetzung innerhalb der Automation ist seit dem Einsatz von Ethernet noch einmal gestiegen.

Industrial Ethernet, bzw. Ethernet-basierte industrielle Kommunikationsprotokolle, weisen in einigen Belangen durchaus Merkmale auf, die so im IT-Einsatzbereich von Ethernet nicht existieren. Vor allem im Bereich der Echtzeitfähigkeit existieren in der Automation Anforderungen, die auf Unternehmensebene so nicht gelten. Für die Kommunikation mit Antrieben per Ethernet werden z.B. Anforderungen im niedrigen zweistelligen Mikrosekunden-Bereich für den Delay und im einstelligen Mikrosekunden-Bereich für Jitter gestellt. Reguläre

Ethernet-Infrastruktur-Geräte (z.B. Switches) wie sie in der Büro- und auch Unternehmenskommunikation eingesetzt werden, können diesen Anforderungen nicht gerecht werden. Organisatorische (z.B. Verkehrsplanung) und technische (spezielle, echtzeitoptimierte Hardware) Maßnahmen helfen dabei, dass Industrial-Ethernet-Geräte genau an diesen Stellen die notwendigen Erweiterungen vornehmen und somit die benötigten Leistungsreserven zur Verfügung stellen.

Eine unvermeidliche Auswirkung durch den verstärkten Einsatz von IT-Technologien in der Automation ist die – unfreiwillige - Integration der Herausforderungen und Probleme, denen sich die IT gegenüber sieht. Betroffen sind unter anderem die Sicherheit von Daten, die Komplexität von Modellen und letztlich die immer schneller und stärker wachsende Komplexität der eingesetzten Gerätschaften und Netzwerke. Diese und weitere Herausforderungen sind in der IT bekannt und es wird ihnen mit verschiedenen Lösungen bzw. Lösungsstrategien begegnet. Lösungen für die genannten Herausforderungen lassen sich jedoch nicht oder nicht durchgängig direkt für die industrielle Automation anwenden. Ihre Anpassung an die Belange und Anforderungen der Automation ist gegenwärtig und zukünftig eine Aufgabe, die Wissenschaftler, Hersteller, Integratoren und Anlagenbetreiber beschäftigt und beschäftigen wird.

2 Ausgewählte Anwendungsfälle

Die Arbeit baut in wesentlichen Punkten auf konkreten Anwendungsfällen auf. An vielen Stellen werden die hier beschriebenen Anwendungsfälle benutzt, um Designentscheidungen nachvollziehbar zu machen, Modelle zu instanziieren, aber vor allem auch, um Praxisnähe und Relevanz nachzuweisen. Die Tiefe, in der einzelne Anwendungsfälle im Verlauf der Arbeit behandelt werden, wertet nicht ihre Wichtigkeit in der Praxis der Automation. Vielmehr sind sie möglichst vielfältig ausgelegt, um sowohl automationsspezifische Sachverhalte, wie auch Sachverhalte, die ihren Ursprung im klassischen IT-Umfeld haben, abdecken zu können. Teilweise bauen Anwendungsfälle aufeinander auf oder teilen sich Aspekte.

Die folgende Beschreibung der Anwendungsfälle dient zunächst der Einführung. Auf notwendige Details wird in den entsprechenden Abschnitten in Abschnitt 6 eingegangen. Die Anwendungsfälle sind über den gesamten Bearbeitungszeitraum der Thematik entstanden und erweitert worden. In Diskussionen mit Kollegen und Fachleuten wurden Details ausgeprägt und beeinflusst, so dass die Anwendungsfälle in ihrer vorliegenden Form zwar keineswegs die gesamte Automation repräsentieren, aber dennoch eine interessante und hinreichend relevante Untermenge von aktuellen und eventuellen zukünftigen Herausforderungen in der Branche darstellen.

2.1 Versions-Management

Geräte, Komponenten, aber vor allem auch Software und Firmware sind in aller Regel versioniert. Grundsätzlich ist das sowohl in der IT als auch in der Automation [4] der Fall. Die Eigenheiten der Automation (vgl. 1.3.1) bringen es mit sich, dass das Versions-Management aktuell eine große Aufgabe darstellt [5], die, wenn überhaupt, nur für spezifische Installationen oder für genau einen Hersteller bzw. eine Produkt-Serie gelöst ist. Allgemeingültige und auf beliebige andere Anlagen übertragbare Ansätze zum anlagenweiten Verwalten von Versionen sind zurzeit nicht bekannt. In der Regel wird einem Versions- und daran anschließend einem Patch-Management nicht oder nicht durchgängig nachgekommen [5]. Im Hinblick auf die große Anzahl an Komponenten und Herstellern ist eine solche Vorgehensweise durchaus nachvollziehbar. Lange war es in der Automation, solange keine expliziten Funktionsfehler vorlagen, auch kaum notwendig auf neuere Versionen zurückzugreifen. Durch kürzere Produktzyklen, steigende Geräte-Komplexitäten und nicht zuletzt durch die zunehmende Vernetzung von Automation-IT und Enterprise-IT wandelt sich die Wahrnehmung der Fachleute und die Aktualisierung von Soft- und Firmware gewinnt an Bedeutung. Unerheblich ist dabei, ob durch Versionen Funktionalitäten und Funktionsumfänge erweitert, beeinflusst oder bestehende Funktionen korrigiert werden.

Die Frage, warum einem aktiven Versions-Management nicht nachgekommen wird, lässt sich nicht pauschal mit großer Heterogenität der Branche oder mit dem erst neuen Bedarf beantworten. Es gibt auch eine Reihe von technischen oder regulatorischen Bedingungen, die das einfache Einspielen von neuen Versionen beeinflussen. Es kann zum Beispiel in besonders kritischen Einsatzgebieten von Automatisierungstechnik notwendig werden, dass nach einer signifikanten Änderung (z.B. Software- oder Firmware-Wechsel) an Teilen der Anlage eine neue Abnahme erforderlich wird.

Auf der technischen Seite kann die simple Ermittlung von aktuellen Firmware-Ständen aller Geräte in einer Anlage einen immensen

Aufwand darstellen, wenn man in Betracht zieht, dass die Anzahl der mit pro Installation verbauten Geräte ohne weiteres mehrere hundert umfassen kann und ein direktes Zugreifen auf gegenwärtige Versionsstände nicht oder nicht direkt möglich ist.

Der an dieser Stelle beschriebene Anwendungsfall befasst sich in erster Linie mit technischen Aspekten des Firmware-Managements bei vernetzten Geräten in einer automatisierungstechnischen Anlage. Eine stark vereinfachte Schrittfolge für einen Soll-Ist-Abgleich von Firmware-Ständen wird in Abbildung 3 dargestellt.

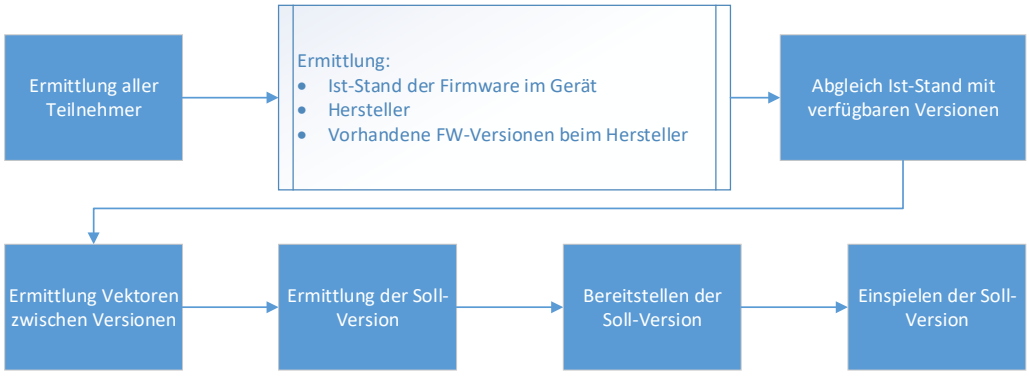

Abbildung 3: Flussdarstellung für Versions-Management, vereinfacht

Zunächst muss ermittelt werden, welche Geräteinstanzen erreichbar sind. Industrielle Kommunikationsprotokolle bieten dafür in aller Regel einen speziellen Dienst zur grundlegenden Geräte-Identifikation an. Als Ergebnis wird eine Liste der erreichbaren Geräte und der Weg, auf dem mit ihnen kommuniziert werden kann (z.B. Netzwerkadressen), geliefert. Mit diesen Basisinformationen kann – feldbusprotokoll-spezifisch – auf weitere Identitäts- und Versionsinformationen der Geräte zugegriffen werden. In der Regel sind Informationen zu Gerätehersteller und gegenwärtigen Versionsständen in den Identifikationsinformationen enthalten. Über Datenbanken oder Webseiten des jeweiligen Herstellers kann eine Liste aller verfügbaren Versionen bezogen und folglich mit dem Ist-Stand der Geräteinstanz abgeglichen werden. Wie nach dem Soll-Ist-Abgleich der Versionsstände weiter verfahren wird, ist abhängig vom Betriebskontext der Geräte. Eher untypisch ist, dass neue Versionen direkt eingespielt werden. Vielmehr

ist davon auszugehen, dass eine eingehende Prüfung der Funktionsänderungen bzw. Korrekturen, die durch das Aufspielen einer neuen Softwareversion hervorgerufen werden, auf Seiten des Anwenders erfolgen muss. Dieser Bewertungsprozess kann mit erheblichen Aufwänden verbunden sein, muss aber in Kauf genommen werden, da das Einspielen von Versionen, die sich im Betrieb anders als erwartet verhalten, zu Beeinträchtigungen an Prozessen und letztlich auch an Produkten führen kann. Ist eine Entscheidung zu Gunsten einer bestimmten Version getroffen, kann diese bereitgestellt und schließlich eingespielt werden.

Die Ermittlung der Änderungen zwischen Versionen und das eigentliche Einspielen einer neuen Softwareversion in ein Feldgerät sind gegenwärtig kritisch, da es weder von Feldbus-Organisationen noch von Herstellern einheitliche Ansätze oder gar Regelungen gibt. Für das Einspielen von neuen Softwareversionen in eine Vielzahl von Geräten führt das im ungünstigsten Fall dazu, dass für n Geräte n Methoden benötigt werden. Für diese beiden gegenwärtig vollkommen offenen Punkte – Ermittlung der Änderungen und Einspielen der aktualisierten Firmware - werden deshalb im Verlauf der Arbeit nur Vorschläge für eine Umsetzung unterbreitet, jedoch nicht technisch umgesetzt. Dafür notwendige Richtlinien müssen Gegenstand der Bemühungen entsprechender Arbeitskreise und Gremien sein.

2.2 Topologie-Erkennung

Komponenten in einem Netz, egal ob es sich um ein Automatisierungsnetz, Firmennetzwerk, Heimnetzwerk oder sonstiges handelt, sind miteinander verbunden. Technologisch kann das heutzutage auf unterschiedlichste Art und Weise passieren, je nach Anwendungsbereich, meist oder zunehmend jedoch Ethernet basiert. Die Struktur, welche die verbundenen Geräte aufspannen - also welches Gerät mit welchem verbunden ist - wird dabei im Allgemeinen als Netz-Topologie oder einfach als Topologie bezeichnet. Prinzipiell sind verschiedene Topologien bekannt: Ring, Bus, Linie, Stern, Vermascht,

Vollvermascht aber auch Mischformen. Welche der genannten Topologien bevorzugt wird, hängt von verschiedenen Faktoren ab, beispielsweise dem Anwendungsgebiet (Automatisierungsnetz, Büronetz, Backbone etc.).

Die genaue Topologie eines betrachteten Netzes zu kennen, ist für eine Reihe von Szenarien unbedingt notwendig. Eine Fehlersuche in umfangreichen Netzwerken wird ohne Kenntnis der Topologie schwer, wenn nicht sogar unmöglich. Die Planung von Bandbreiten und benötigten Ressourcen, z.B. für zu priorisierende Echtzeitanwendungen, wird ohne Kenntnis der Topologie ebenfalls unmöglich. Beide Beispiele stellen nur einen kleinen Auszug aus den vielfältigen Bedarfsszenarien dar.

Die gegenwärtige Topologie eines Netzes lässt sich grundlegend auf drei Arten ableiten:

- das manuelle, sozusagen offline, Ablesen der Topologie aus dem Ist-Stand,
- das Entnehmen der Topologie aus Plänen, die zum Design, oder Re-Design, des Netzes angefertigt wurden und
- die gegenwärtige Topologie kann auch online aus dem Netz selbst ermittelt werden.

Keiner der Ansätze ist dabei frei von Herausforderungen. Die „offline Ablesen"-Herangehensweise ist für komplexe Netze entweder grundsätzlich unmöglich, weil Teile unter Umständen nicht zugänglich sind, oder mit großem Aufwand verbunden. Um ein möglichst vollständiges Bild zu erhalten, müssen nämlich das Abschreiten der gesamten Installation und die manuelle Erfassung jeder Komponente, sowohl aktiv als auch passiv, und jeder Verbindung zwischen Komponenten in Betracht gezogen werden. Die Ermittlung der Topologie aus vorhandenen Topologie-Plänen ist prinzipiell der genaueste und schnellste Weg, da nur so wirklich alle aktiven und vor allem auch passiven Teilnehmer am Netz identifiziert und ihren Nachbarn zugeordnet werden können. In einer ideal betriebenen Automatisierungsanlage aber auch in einem Firmennetz werden exakte offline-Netzpläne immer vorliegen und vor allem auch den aktuellen Stand des

Netzes beinhalten. Die Realität zeichnet jedoch ein anderes Bild. Es werden immer wieder kleinere Änderungen am Netz vorgenommen, die eben nicht in die Planungsdaten zurückfließen, spontan Knoten ausgetauscht und durch nicht identische ersetzt oder (Sub-)Netze erweitert. Letztlich könnte man sich nicht auf vorliegende Pläne verlassen. Einzige Alternative ist, die Topologie aus dem laufenden Netz heraus zu ermitteln. Dieser Vorgang wird Topologie-Erkennung genannt. Im Weiteren bezeichnet Topologie-Erkennung immer die online-Ermittlung der Netzstruktur aus den teilnehmenden Komponenten, welche auch als Knoten bezeichnet werden.

Im IT-Umfeld befassen sich Forschergruppen und Unternehmen seit Jahren mit dem Problem der Topologie-Erkennung. Dabei wird häufig in Schichten gegliedert, das Spektrum reicht von der Link-Layer-Topologie bis zur Overlay-Topologie. [6] bietet einen guten Überblick über die Bemühungen verschiedener Forschergruppen in diesem Bereich. Die Ansätze unterscheiden sich maßgeblich im Hinblick auf geforderte Eigenschaften der Knoten, Netzgliederung (Subnetze, VLAN) und Vollständigkeit bzw. Vorhandensein bestimmter Informationen. Die Ansätze können grob in zwei Klassen unterteilt werden: Topologie-Erkennung mit Netzwerkunterstützung und Topologie-Erkennung ohne Netzwerkunterstützung. Die zuletzt genannte geht davon aus, dass die Knoten nicht selbst Informationen bereithalten, die zu einer Topologie-Erkennung genutzt werden können. Um dennoch die Struktur des Netzes abzuleiten, ist es bei diesem Ansatz nötig, auf einer Anzahl Knoten (dabei gilt je *mehr desto besser*) eine spezielle Software zu installieren. Details zu diesem Ansatz werden unter anderem in [7] dargestellt. Für die industrielle Automation ist dieser Ansatz damit ungeeignet, da die Vielzahl der Knoten in diesem Bereich weder genügend Ressourcen für eine solche Software zur Verfügung stellen kann, noch Softwareinstallation im üblichen Sinne überhaupt möglich ist.

Auch die Topologie-Erkennung mit Netzwerkunterstützung stellt verschiedene Ansprüche an die Knoten. Je nach konkretem Algorithmus werden unterschiedliche Informationen und Informationsdichten

gefordert, die von den Knoten per SNMP (Simple Network Management Protokoll [8]) abgerufen werden um daraus eine Topologie zu errechnen. SNMP ist de-facto Standard für das Netzwerk-Management auch in der Automation [9], dort wird er von einigen Organisationen sogar vorgeschrieben und die Implementierung ist zwingend. Welche Informationen per SNMP bereitgestellt werden, ist jedoch im Rahmen vorgegebener Informationssätze frei. Das bedeutet, wenn ein Gerätehersteller einen für den jeweiligen Topologie-Erkennungsalgorithmus notwendigen Teil des Informationshaushaltes nicht vorgesehen hat, scheidet dieses Gerät aus bzw. wird ein unzuverlässiger Teil der Topologie. Je nach Algorithmus und geforderten Informationen ist eine gewisse Anzahl von „unzuverlässigen" Knoten tolerierbar. Technische Details sind unter anderem in [6] beschrieben. Allen Verfahren zur Topologie-Erkennung ist gemein, dass passive (Kabel, Stecker) und nicht-intelligente (Hubs, nichtkooperative) Komponenten weitgehend transparent bleiben, also wahrscheinlich nicht als expliziter Teil der Topologie ausgewiesen werden können.

Grundsätzlich lassen sich genannten Ansätze aus der IT auch auf den Bereich der Automation anwenden, vor allem, da die in Absatz 1.3.1 beschriebenen Besonderheiten der Automation an Prägnanz verlieren, je weiter man sich von der Feldebene weg hin zur Unternehmensebene bewegt. Für die vorliegende Arbeit kann bemerkt werden, dass, sollten die im Rahmen dieser Arbeit benutzten Informationen nicht verfügbar sein, die genannten Ansätze zur Topologie-Erkennung mit Netzwerkunterstützung aus der IT eine Komplettierung darstellen können. Sie sind aber selbst nicht Gegenstand der Betrachtungen.

Ziel des Anwendungsfalls Topologie-Erkennung ist es, eine Link-Layer-Topologie mit Netzwerkunterstützung aus bestehendem Industrial Ethernet mit automatisierungstechnischen Geräten abzuleiten. Dabei wird folgendes generelles Vorgehen angenommen, Abbildung 4:

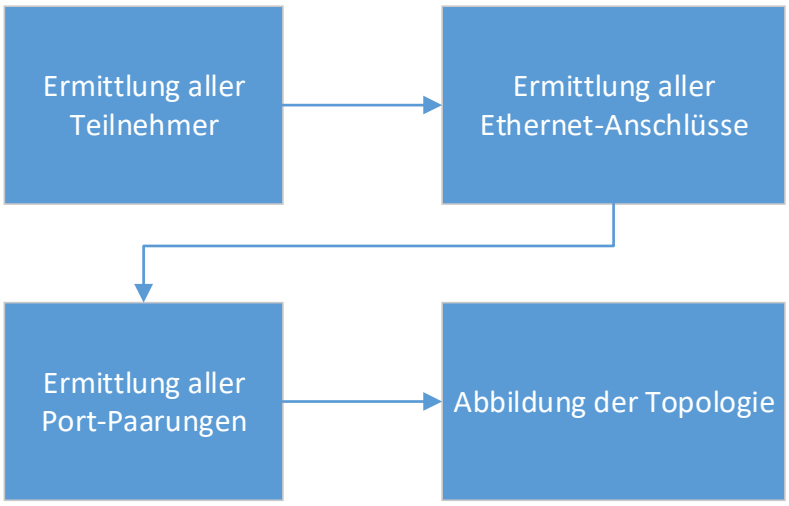

Abbildung 4: Generelles Vorgehen Link Layer Topologie

Zunächst müssen alle Komponenten/Knoten im Netz identifiziert werden. Darauf folgend wird ermittelt, wie viele (Ethernet-)Anschlüsse ein Knoten vereint und somit die maximale Anzahl an direkten Nachbarn abgebildet. Abschließend wird zu jedem (Ethernet-)Anschluss der konkrete Nachbaranschluss eines vernetzten Knotens ermittelt und die Knoten selbst anhand der benachbarten Anschlüsse in Beziehung gesetzt.

Für sich betrachtet ist der Anwendungsfall Topologie-Erkennung auf den ersten Blick von geringerer Komplexität als etwa das Versions-Management oder das Alarm-Handling, allerdings bildet er einen wichtigen Informationsgrundstock und ermöglicht weitere Anwendungsfälle und Szenarien überhaupt erst. Die Behebung eines Verbindungsdefektes zwischen zwei Automatisierungsgeräten würde ohne Kenntnis der genauen Link-Layer-Topologie unnötig erschwert. Weiterhin ist es so, dass der Anwendungsfall eine Möglichkeit aufzeigen soll, die weg von der gegenwärtig in der industriellen Automation üblichen Praxis der proprietären, herstellerspezifisch beschränkten Topologie-Werkzeuge hin zu Topologie-Repräsentationen mittels allgemeingültiger, offener und erweiterbarer Modelle führt. Nur so kann langfristig ein durchgängiges Automatisierungssystem-Management erreicht werden.

2.3 Alarm Handling

Alarme werden in der Regel bei Zustandsübergängen ausgelöst. Der Alarm-Begriff ist im technischen Umfeld unterschiedlich belegt. Für den Kontext dieser Arbeit soll ein Alarm ein Ereignis sein, welches eine Änderung am Systemzustand anzeigt. Für das Automatisierungsumfeld kann man grundsätzlich zwei Arten von Alarmen unterscheiden:

- Prozessalarme, die direkten Bezug auf einen verfahrenstechnischen Prozess haben. Sie informieren beim Unter- oder Überschreiten von vorgegebenen Füllständen, Temperaturen, Drücken und so weiter. Prozessalarme werden auf Pro-Anwendungs-Ebene konfiguriert (z.B. erlaubter Maximalwert einer Temperatur).
- Systemalarme sind all die Alarme, die ausgelöst werden, sobald eine Zustandsänderung an den Komponenten verzeichnet wird, die den eigentlichen Prozess steuern. System-Alarme sind in aller Regel vorgegeben.

Der Neustart eines Gerätes, die Aktivierung/Deaktivierung von Geräten und Geräteteilen sind Beispiele für System-Alarme. Geräteteile, die im Betrieb entfernt werden, lösen, sofern ordnungsgemäß in Betrieb genommen, z.B. in den meisten Automatisierungssystemen einen (System-)Alarm aus. Welche Alarme in einer Automatisierungsanlage letztlich genau ausgelöst werden, hängt mit den konkret eingesetzten Technologien zusammen. Ob und wie aufgetretene Alarme behandelt werden, bzw. welche weiteren Schlüsse aus einem Alarm gezogen werden, steht dem Anwender frei.

Prozessalarme werden im Verlauf dieser Arbeit eine untergeordnete Rolle spielen. Lediglich Abschnitt 4.4 wird dieses Thema noch einmal kurz aufgreifen, da es sich bei der dort diskutierten Technologie (OPC UA) um den einzigen Vertreter aus der Prozessindustrie handelt und dort Prozessalarme durchaus verankert sind. Etablierte Systeme zum Führen von Prozessen sind bestens geeignet um mit Prozessalarmen

umzugehen. Alarm wird daher im Weiteren synonym für System-Alarm verwendet.

Die Notwendigkeit für die Integration von Alarmen in ein System-Management ergibt sich vor allem aus der Vielfalt an Technologien und Standards, die in der Automation zum Einsatz kommen. Nicht in allen Ethernet-basierten Feldbussen ist ein Netzwerk-Management mit entsprechenden Nachrichten-Funktionalitäten vorgeschrieben. Die Integration von Alarmen kann so unter Umständen die einzige Möglichkeit bieten, über Veränderungen am Zustand der Infrastruktur des Systems informiert zu werden. Auf der anderen Seite sind im System-Management evtl. Informationen abgebildet, die in einem klassischen Prozessführungssystem fehlen, welche aber zusammen mit einem aufgetretenen Alarm an Semantik gewinnen. Weiterhin ist der von Automatisierungsgeräten vorgehaltene Speicher für aufgetretene Alarme häufig begrenzt, eine Alarmhistorie kann so verloren gehen. Dies kann vor allem dann der Fall sein, wenn in einem kurzen Zeitraum viele Alarme auflaufen (Meldeschwall). Nicht zuletzt ist die Zuordnung von aufgetretenem Alarm zu Gerät oder Geräteteil, Netzwerk oder ganzer Anlage nicht trivial. Die Integration von Alarmen in ein System-Management kann auch hier helfen, indem sie etwa ein Gerät mit dem von ihm in der Vergangenheit ausgelösten Alarm automatisch assoziiert. Dieser Anwendungsfall soll die übliche Alarmbehandlung im Rahmen der Automatisierungsapplikation nicht ablösen, sondern lediglich um Aspekte des System-Managements erweitern. Vor allem im Hinblick auf die Problematik des Quittierens von Alarmen ist die vollständige Verlagerung des Alarm-Handlings in ein Management-System kritisch zu betrachten.

Dargestellt in Abbildung 5 ist das grundsätzliche Vorgehen, um auftretende Alarme im Management-System zu registrieren. Ein aufgetretener Alarm wird mittels eines Logbuches, welches gleichartige Alarme gruppiert, im System abgebildet und zugeordnet. Damit der Alarm selbst auch eine systemische Bedeutung hat, wird er mit allen Entitäten, mit denen er direkt in Relation steht, assoziiert.

Voraussetzung dafür ist, dass alle assoziierten Entitäten selbst eine Abbildung im Modell des Management-Systems aufweisen.

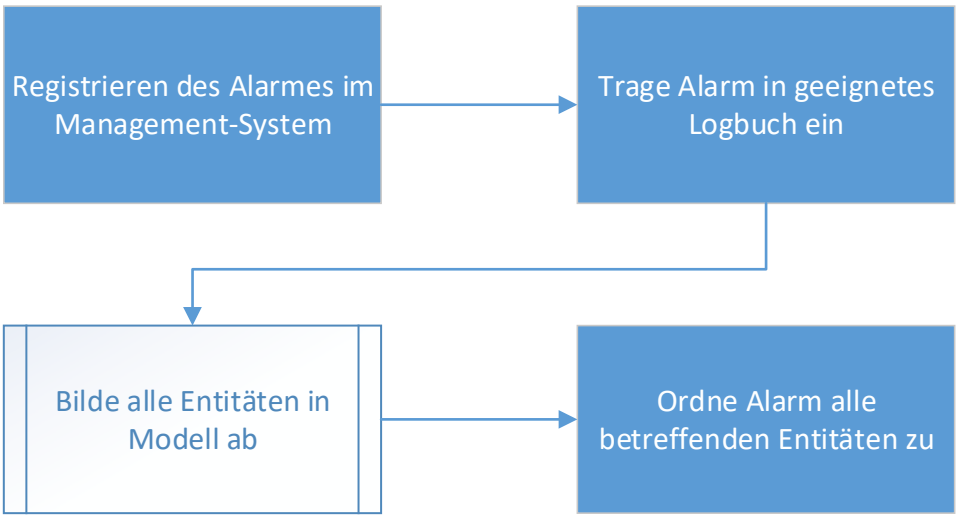

Abbildung 5: Generelles Vorgehen Alarm Handling

Für den Modus der Übermittlung von Alarmen aus einer automatisierungstechnischen Anlage heraus hin zu einem Management-System sind verschiedene Wege denkbar. Es muss jedoch beachtet werden, dass hier kaum ein allgemeingültiges Vorgehen möglich ist, da je nach Protokoll einzelne Wege explizit ausgeschlossen sind oder nur durch „Workarounds" beschritten werden können. Auf die Repräsentation von Alarmen in einem Management-System muss dies jedoch keinen nachhaltigen Einfluss haben. Detaillierte Beschreibungen möglicher Herangehensweisen sind in Abschnitt 6.4 diskutiert.

2.4 Dienste-Management

Der Begriff Dienste-Management wird seiner englischen Entsprechung – Service Management - nach in vielen Bereichen unterschiedlich verwendet. Im Umfeld des ITSM (IT Service Management) hat er Bezug zu umfangreichen Teilen von Geschäftsprozessen wie sie heute in vielen Unternehmen anzutreffen sind. Verschiedene Branchen vertrauen dabei auf unterschiedliche Rahmenwerke und de-facto Standards. Als wichtige Vertreter seien an dieser Stelle ITIL (IT Infrastructure Library) und eTOM (enhanced Telecom Operations Map) genannt. Ein knapper Überblick

über beide, aber vor allem die Beziehungen zwischen ihnen, wird in [10] geliefert. Sowohl ITIL wie auch eTOM berühren zwar auch technische Aspekte von Diensten, wenn auch in sehr unterschiedlichem Umfang, sind aber nicht grundsätzlich darauf ausgerichtet diese unmittelbar zu steuern. Die Ausrichtung gilt vielmehr Diensten im Sinne von Dienstleistungen und Diensterbringung, mit unterschiedlichem Fokus, also Dienste im Rahmen von Geschäftsprozessen. Speziell ITIL versucht eine Reihe von Best Practice Anweisungen zur Durchsetzung von ITSM zur Verfügung zu stellen, jedoch keine Technologien zu spezifizieren. eTOM zielt auf die Datenintegration zwischen Geschäftsprozessen ab, um eine durchgängige Diensterbringung überhaupt zu ermöglichen. Das System-Management hat durchaus Berührungspunkte mit ITIL und eTOM. Vor allem, wenn es z.B. um Abrechnung (billing), die Erfüllung von SLAs (Service Level Agreement) o.A. geht.

Es gilt zunächst die im Rahmen dieser Arbeit als Gegenstand eines Managements betrachteten Dienste genauer zu spezifizieren und bezüglich der Belange, die für die industrielle Automation von besonderer Bedeutung sind, einzugrenzen. Durch den Fokus auf betriebswirtschaftliche Aspekte eignen sich die ITIL Service-Definition [11] wie auch der für eTOM geprägte Servicebegriff [10] nicht für den technischen Kontext. Auch wenn das Themenfeld Service Oriented Architecture (SoA) im Rahmen dieser Arbeit nicht dediziert betrachtet wird, erscheint die Definition aus [12] weitestgehend geeignet:

„A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description."

Diese Definition deckt alle Dienste, die in diesem Anwendungsfall von Bedeutung sind, ab. Spezielles Interesse gilt den Diensten, die von Automatisierungsgeräten angeboten werden, vor allem denen, die nicht unbedingt essentiell für die Funktion des jeweiligen Gerätes sind. Das sind all die Dienste, die etwa weitergehende Konfigurationen ermöglichen, Betriebszustände auf einer Webseite anzeigen oder

allgemeine Server Dienste. Dabei kann es sich bspw. um Konsolenzugänge (ssh, telnet) und integrierte Webserver oder FTP-Server handeln. Den genannten Diensten ist gemein, dass sie aus Sicht einer Anlage nicht unbedingt notwendig sind, den Komfort und die Flexibilität aber erhöhen können.

Auch in diesem Anwendungsfall muss die Frage nach der Notwendigkeit und dem Nutzen beantwortet werden. Zunächst ist es für den Betreiber einer Anlage in der Regel wünschenswert über alle Gegebenheiten, in diesem Fall von Geräten angebotene Dienste, die Hoheit zu besitzen. Eben auch dann, wenn man diese Dienste nicht für den Betrieb im Tagesgeschäft benötigt, aber beispielsweise für regelmäßige Audits, Anlagenwartungen oder Fehlerdiagnosen heranziehen möchte. Die Diensthoheit ist auch für den sicheren – im Sinne informationstechnischer Sicherheit (Security) – Betrieb einer Anlage von unbedingter Bedeutung. Unkenntnis über die in der eigenen Anlage aktuell betriebenen Dienste und Zugänge kann dazu führen, dass Sicherheitsverletzungen der Weg bereitet wird. Auf der anderen Seite können fehlerhaft oder unerwartet konfigurierte Dienste direkt dazu führen, dass in der jeweiligen industriellen Applikation unerwünschtes oder unerwartetes Verhalten auftritt. Dienstkenntnis ist also für den reibungslosen Betrieb generell notwendig.

Die Hoheit über das eigene Geräteportfolio zu besitzen, ist unumstritten von großer Bedeutung. Es stellt sich allerdings die Frage, was eine prinzipielle Hoheit über Geräte in einer hochgradig heterogenen Umgebung einem Betreiber oder Anwender praktisch nutzt, wenn sie sich vorwiegend auf einzelne Komponenten, nicht aber auf ganze (Teil-)Anlagen oder Systeme bezieht. Für die in Absatz 1.3.1 bereits beschriebenen Anlagengrößen lässt sich die Antwort leicht ableiten. Generell kann in einem solchen Fall nicht mehr von Handhabbarkeit ausgegangen werden. Das Management von Diensten mithilfe eines System-Management-Ansatzes kann hier die Lösung bringen.

Der in Abbildung 6 dargestellte Ablauf beim Dienste-Management geht zunächst wieder davon aus, dass alle Teilnehmer, die Dienste potentiell anbieten können, erst einmal erkannt werden müssen. Dies geschieht

analog zu den vorhergehenden Anwendungsfällen. Für jeden aufgefundenen Teilnehmer muss nur ermittelt werden, welche Dienste dieser prinzipiell anbieten kann, welchen Status der jeweilige Dienst hat (aktiviert, deaktiviert, pausiert etc.) und nicht zuletzt welche Parameter aktuell eingestellt sind bzw. vorgehalten werden. Das Ändern von Konfigurationen und Status würde in weiteren Schritten erfolgen.

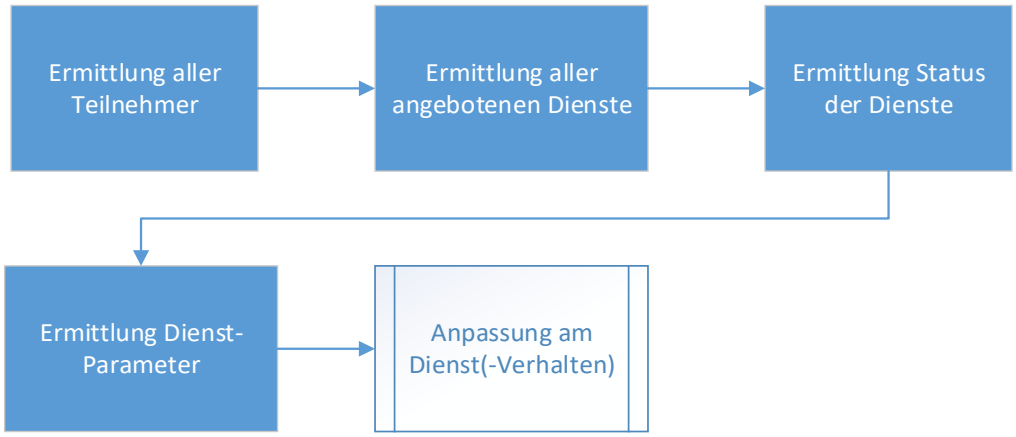

Abbildung 6 Generelles Vorgehen Dienste-Management

Neben den offensichtlichen administrativen Herausforderungen bei der Verwaltung von heterogenen Diensten in großen heterogenen Netzwerken, dem auch teilweise mit schon bestehenden Netzwerk-Mangement-Ansätzen begegnet werden kann, gibt es noch eine weitere Problematik, die in realen Installationen immer wieder auftritt. Hierbei handelt es sich um den Vorgang der Diensterkennung selbst. Es gibt zwar Ansätze, wie etwa SLP (Service Location Protocol), die das Vorhandensein bestimmter Dienste auf bestimmten Knoten bekannt machen, diese haben aber in der Automation bis jetzt keinerlei Bedeutung und sind auch generell nicht durchgängig für jeden Dienst implementiert. Fraglich bleibt, wie selbst im Falle, dass sich beispielsweise SLP in der Automation etablieren könnte, eine Durchgängigkeit herbeigeführt werden soll. Der entgegengesetzte Ansatz, das aktive Suchen nach Diensten (Netzwerkscan), ist in der Praxis kaum anwendbar (Netzwerkbelastung, unvorhersagbares Verhalten der Teilnehmer) und fehlerbelastet (deaktivierte, unbekannte etc. Dienste werden in aller Regel gar nicht erkannt) und bietet somit keine

Alternative. In der Automation gäbe es eine Reihe von Ansätzen, die hier weiterhelfen können [13]. Merkmalsleisten [14], elektronische Typenschilder [15], Security Data Sheet [16] sind bekannte, jedoch nicht durchgängig umgesetzte Vertreter. Letzlich müssen sich Geräteher-steller auf einen Ansatz einigen und ihn umsetzen. Die Verwaltung von Diensten, ob per System-Management oder Einzelzugriff, ist ohne Kenntnis ihrer Existenz kaum lückenlos möglich. Andererseits bietet das System-Management hier Potential, nicht nur bei der Verwaltung, sondern auch als Instrument zur Bekanntmachung von Diensten.

2.5 Schlussfolgerungen aus den Anwendungsfällen

Mit den gewählten Anwendungsfällen wurde versucht, ein breites Feld von Themen zu adressieren, die in Bezug auf Management-Aufgaben heute in der Automatisierungstechnik nicht, oder - die Interoperabilität stark einschränkend - nur proprietär gelöst sind, aber dennoch gewisse Zusammenhänge aufweisen. Es ist naheliegend, dass nicht alle Aspekte betrachtet werden können. Management-Aufgaben, die direkt und vordergründig die Datensicherheit betreffen, sind ebenso außer Acht geblieben wie etwa Betrachtungen zur allgemeinen Leistungsevaluation von Systemen, Diagnosen, Fehlerbehebungen, Ressourcen-Manage-ment. Diese Liste ließe sich noch fortführen und bietet Raum für weitere Arbeiten. Keinesfalls soll mit der Auswahl der Anwendungsfälle die Wichtigkeit oder Priorität einzelner Management-Aufgaben gewertet werden. Eine solche Bewertung wäre im Allgemeinen auch kaum möglich, da sie von Anwendungsdomäne, konkreten Installationen mit speziellen Anforderungen und letztlich von verwendeten Komponenten abhängt.

Die gewählten Anwendungsfälle eignen sich jedoch gut, um Herausforderungen in Bezug auf Netzwerk- und System-Management allgemein, wie auch spezielle Anforderungen, die durch die Automation bzw. die eingesetzten Technologien gegenüber einer System-Manage-ment-Lösung entstehen, herauszuarbeiten. Alle im Folgenden kurz

beschriebenen Herausforderungen ergeben sich aus den Anwendungsfällen und beziehen sich ausdrücklich auf Aspekte des System-Managements. In Abschnitt 3.2 werden unter anderem auf Basis der herausgearbeiteten Herausforderungen Anforderungen an System-Management-Ansätze abgeleitet.

Vielfalt bei Geräten, Herstellern, eingesetzten (Feldbus-)Protokollen und Schnittstellen

Belange, die im eigentlichen Sinne eine Management-Aufgabe darstellen (Konfigurationsverwaltung, Informationszugriff, Fehlermanagement etc.), sind in der Automation als Einzelimplementierung umgesetzt. Dabei können die Einzelimplementierungen in diesem Kontext in die drei Kategorien pro Anlage, pro Hersteller und pro Feldbus eingeteilt werden. Ansätze, die pro Anlage umgesetzt werden, bieten in der Regel nur beschränkte Unterstützung für gegenwärtig oder geplant nicht vorhandene Management-Aufgaben. Mit Blick auf die Erweiterbarkeit einer Management-Lösung ist dies ein problematischer Sachverhalt. Herstellerspezifische Implementierungen auf der anderen Seite bieten kaum Unterstützung für herstellerübergreifende Management-Aufgaben, was in Multi-Vendor-Umgebungen dazu führt, dass für gleiche bzw. ähnliche Management-Aufgaben unterschiedliche Werkzeuge zum Einsatz kommen.

Durch die starke Heterogenität auf Seiten der Automatisierungs-Hardware haben sich eine Vielzahl von unterschiedlichen Schnittstellen und Methoden für prinzipiell ähnliche Aufgaben herausgebildet (vgl. 1.3.1).

Komplexität der Geräte, Anlagen und Anforderungen an die Automation

Steigende Geräteanzahl und Netzwerkkomplexität gehören generell zu den wesentlichen Gründen, sich für ein komplexes Netzwerk-Management und gegen Lösungen zum Verwalten, Beobachten und Steuern einzelner Aspekte zu entscheiden [17], [18], [19], [20], [21] – auch in der Automation. Eine zunehmende Komplexität der Geräte führt

im Allgemeinen auch dazu, dass die Anzahl an les- und konfigurierbaren Werten steigt. Die Verwaltung dieser wachsenden Anzahl an Werten erweist sich schnell als aufwändig und fehleranfällig bezüglich Eingabefehlern, vor allem dann, wenn mit großen Geräte-Pools gearbeitet wird. Ein ausschließlich auf dem Lesen und Schreiben von einzelnen Werten basierendes Management kann die Leistungsfähigkeit des Managements einschränken.

Die sich vor dem Hintergrund der aktuellen Diskussionen bezüglich Industrie 4.0 und Cyber Physical Systems ändernde Wahrnehmung der Automation wird in der Zukunft noch einmal gestiegene Anforderungen an das Netzwerk- und System-Management mit sich bringen. Sich flexibel re-konfigurierende Fertigungsprozesse werden nur möglich sein, wenn auch die ausführende Infrastruktur, also alle Geräte- und Softwarekomponenten, die ein Automatisierungssystem prägen, mit leistungsfähigen und zukunftsgeeigneten Ansätzen verwaltet wird.

Informations-Repräsentation, -Semantik und –Relation

Unabhängig von der Anwendungsdomäne - IT oder Automation -, das System- und Netzwerk-Management lebt von der allgemeinverständlichen und aufgabenorientierten Repräsentation von Informationen. Im Umkehrschluss bedeutet das, Informationen müssen von einem Ansatz geeignet dargestellt und gehalten werden können. Einfache Werte sind beispielsweise ohne Datentypen, Bedeutung, Kontext und evtl. Relation zu anderen Werten aus Sicht eines Management-Systems, aber auch aus Sicht des Anwenders, weitestgehend nutzlos. Im Zusammenhang mit dem beschriebenen Komplexitätszuwachs in der Automation steigt auch die Anforderung an die Darstellung des Informationshaushaltes. Es müssen regelmäßig neue Informationen abgebildet, die Informationsrepräsentation dafür entsprechend erweitert werden.

Integration von bestehenden Ansätzen

Netzwerk-Management ist in der Automation kein vollkommen unbekanntes Feld. Zum einen kommen vereinzelt schon Netzwerk-

Management-Protokolle zum Einsatz, zum anderen werden Management-Aufgaben häufig mit Einzellösungen umgesetzt. Es ist mittelfristig ausgeschlossen, dass in der Automation auf gegenwärtig eingesetzte Management-Technologien zu Gunsten von neueren vollkommen verzichtet wird. Betriebsbewährtheit und Laufzeiten von Installationen spielen hier eine wesentliche Rolle.

Akzeptanz von neuen Technologien in der Automation

Neue Ansätze werden in der Automation nicht so leicht aufgegriffen wie in anderen Technologiebereichen. Jegliche Management-Ansätze müssen also neben ihrer technischen und theoretischen Eignung auch bezüglich ihrer subjektiven Eignung bewertet werden.

Verbesserte Integration von Belangen der Automation in die IT

Ethernet, allgemein IT-Technologien, finden immer breiteren Einsatz in der Automation. Das betrifft aber gegenwärtig vor allem Technologien, nicht jedoch Verwaltungsstrukturen und Werkzeuge. Vereinheitlichung der Methoden zum Netzwerk-Management ist ein wesentlicher Punkt, damit Automation und IT auch auf administrativer Ebene eine bessere Integration erfahren können.

Soll-Ist-, Online-Offline-, Engineering-Live-Abgleich

Netzwerk-Management und System-Management haben in der Regel die Intention den gegenwärtigen Zustand des Systems, das sie kontrollieren, darzustellen und im Rahmen ihrer Möglichkeiten zu steuern. In der Automation kommt dazu noch ein weiterer wesentlicher Punkt, nämlich der Abgleich zwischen geplanten Daten (vgl. Abschnitt 1.3.1) und tatsächlichen sowie, in direktem Zusammenhang damit, auch die Darstellung von gegenwärtig nicht aktuell abrufbaren Informationen, z.B. von Geräten, die vorhanden sind, sich jedoch aktuell in einem deaktivierten Zustand befinden.

Vor allem für die Unterstützung eines Operators oder Administrators während des Anlaufes von (Teil-)Anlagen ist, abseits von proprietären

Lösungen, der Soll-Ist-Vergleich auf verschiedenen Ebenen von Bedeutung. Dafür müssen auch Informationen repräsentierbar sein, die sich auf gegenwärtig nicht aktive Komponenten beziehen.

Ereignisse aus dem Feld, welche unmittelbar behandelt werden müssen

Neben dem Lesen und Setzen von Daten durch eine Management-Lösung kann es, gerade in der Automation, vorkommen, dass in einer gemanagten Komponente ein neuer Zustand auftritt, über den etwa der Operator dringend informiert werden muss. In einem solchen Fall kann nicht gewartet werden, bis Informationen mittels Management ausgelesen werden, sondern es muss, ausgelöst von der betreffenden Komponente, eine direkte Nachricht an den Operator bzw. an das Management-System ausgelöst werden, damit eine zeitnahe Behandlung des Zustandes erfolgen kann.

Persistieren von Management-Informationen

Managementrelevante Informationen sind in der Regel Momentaufnahmen. Für die Automation ist es aber die Regel, dass bestimmte Informationen, die im Sinne dieser Arbeit Management-Informationen darstellen, persistiert werden um Abbilder bestimmter zurückliegender Sachverhalte abrufen zu können. Das Persistieren von Informationen ist eng verzahnt mit der Informationsrepräsentation und dem Online-Offline-Abgleich von Informationen.

3 Netzwerk-Management in der Automation

Die Gründe sich für ein explizites Netzwerk-Management zu entscheiden, sind in der Automation wie auch in der klassischen Enterprise IT im Wesentlichen identisch: steigende Netz- und Systemkomplexität. So ist es auch nicht überraschend, dass sich grundlegende Modelle des Netzwerk-Managements in der Automation nicht von denen in der IT unterscheiden. Die vier in diesem Kontext relevanten Modelle sind: Informationsmodell, Kommunikationsmodell, Organisationsmodell und Funktionsmodell (vgl. [22] [17] [23]).

Das Informationsmodell

beschreibt und definiert wie Informationen gehalten und repräsentiert werden, sowie die Relation, in der Informationen zueinander stehen.

Das Kommunikationsmodell

definiert die eigentliche Kommunikation der Daten über eine Kommunikationsinfrastruktur, also die Encodierung der Informationen in PDUs (Packet Data Units), aber auch die angebotenen Dienste, anhand derer auf den Inhalten des Informationsmodells Operationen ausgeführt werden können.

Das Organisationsmodell

definiert die internen Strukturen des Ansatzes, also Prozesse, aber auch Hierarchien zwischen Managern und gemanagten Komponenten.

Das Funktionsmodell

Beschreibt, welche funktionalen Ansprüche an das Management-System gegenüber dem zu managenden System gestellt werden.

Das Funktionsmodell definiert die so genannten SMFAs (System Management Functional Areas), siehe dazu Abschnitt 4.1.

Alle hier kurz beschriebenen Modelle, inklusive SMFA, wurden mit dem ersten durchgängigen Management-Ansatz definiert (Open System Interconnection Management) (Absatz 4.1). Heute ist dieser Ansatz zwar praktisch nur noch sehr eingeschränkt im Einsatz, seine Konzepte sind aber in nahezu allen modernen Netzwerk-Management-Technologien nach wie vor gültig.

Die Tragweite der einzelnen Modelle wird in [23] sehr detailliert erläutert und auch ihre Wichtigkeit für einzelne Aspekte des Netzwerk-Managements bewertet. Während ein Informationsmodell und wesentliche Aspekte des Kommunikationsmodells bei nahezu allen Netzwerk-Management-Ansätzen explizit vorhanden sind, gilt dies nicht für Organisations- und Funktionsmodell. Letztere sind auch bei weit verbreiteten Ansätzen oft nur implizit vorhanden, nicht jedoch explizit spezifiziert. Um als vollwertiger, eigenständiger Netzwerk-Management-Ansatz betrachtet zu werden, bedarf es in der Regel wenigstens eines eigenständigen Informations- und Kommunikationsmodells. Dies gilt insbesondere für diese Arbeit und damit die in Abschnitt 4 im Detail betrachteten Technologien und Ansätze.

An dieser Stelle ist es notwendig drei grundlegende Begriffe, die im Rahmen des Managements von Systemen immer wieder auftauchen, kurz abzugrenzen und ihre Bedeutung und Verwendung im Rahmen dieser Arbeit zu erläutern:

Netzwerk-Management (i) & System-Management (ii)

Diese beiden Begriffe klar abzugrenzen, ist in vielerlei Hinsicht schwierig. Auf einer sehr abstrakten Ebene könnte man sagen, das Netzwerk-Management ist eine Teilmenge des System-Managements, es adressiert bzw. repräsentiert nämlich genau die Aspekte

der Netzwerkinfrastruktur des System-Managements. Im Gegenzug bezieht das System-Management zum Beispiel Aspekte des Software-Managements (Inventar, Versionen etc.) mit ein, prägt Systemgedanken (bezogen auf eine ganzheitliche Sicht auf Computersysteme) weiter aus als das reine Netzwerk-Management und beinhaltet zum Beispiel Aspekte zu Vertragsbedingungen, Fristen etc. Andererseits gibt es auch immer wieder Bemühungen, Aspekte, die das Netzwerk-Management überschreiten, in Netzwerk-Management-Protokolle zu integrieren. Wieder andere [17] stellen heraus, dass Management-Ansätze, die auf ein OSI-Anwendungsschicht-Protokoll zurückgreifen bzw. definieren, klar als System-Management zu werten sind. Eine präzise Abgrenzung zwischen Netzwerk- und System-Management ist somit kaum möglich. Es ist nicht unüblich diese beiden Begriffe beinahe synonym zu verwenden. Sofern im Rahmen dieser Arbeit eine Trennung notwendig ist, wird auf diesen Sachverhalt gesondert hingewiesen. In der Literatur, vor allem in älteren Werken [24], [25] wird noch weiter unterteilt, etwa in Anwendungs- und Dienste-Management, aber auch darauf hingewiesen, dass eben diese Trennung zukünftig kaum noch haltbar sein wird. Als vereinender Begriff wird in diesem Zusammenhang integriertes Management oder eben auch System-Management verwendet.

Netzwerk-Management-System (iii)

Als Netzwerk-Management-System (NMS) wird im Allgemeinen Software bezeichnet, die Management-Aspekte einem (menschlichen) Nutzer präsentiert und mit ihm interagiert. NMS müssen nicht auf ein zugrundeliegendes Kommunikationsmodell beschränkt sein, sondern können sich neben mehreren Netzwerk-Management-Protokollen auch weiterer Schnittstellen bedienen, egal ob Open Source oder proprietär. Die Liste der am Markt verfügbaren NMS ist sehr umfangreich, [26] versucht einen Überblick zu geben und NMS in verschiedene Kategorien einzuordnen. NMS definieren in der Regel kein standardisiertes Informations- und Kommunikationsmodell, sie sind also nicht als Netzwerk- oder System-Management-

Protokolle zu verstehen, sondern nutzen diese lediglich. Sie können unter Berücksichtigung der genannten Aspekte als *Frontend* zu Netzwerk- oder System-Management-Ansätzen verstanden werden. Daraus leitet sich die untergeordnete Rolle für diese Arbeit ab, die sich im Wesentlichen mit dem *Backend* - für die Automation geeigneter Management-Ansätze - befasst.

3.1 Technologiequerschnitt

Aspekte des Netzwerk-Managements werden in der Automation auch gegenwärtig realisiert, es ist keineswegs eine Entwicklung, die ausschließlich in der neuesten Vergangenheit stattfand. Man muss allerdings bisher zwischen zwei wesentlichen Punkten unterscheiden: (i) dem Management der IT-Strukturen innerhalb der Automation und (ii) dem Management der eigentlichen automatisierungstechnischen Anlage. Für den ersten Aspekt ist naheliegend, dass bewährte Management-Ansätze aus der Unternehmensebene wiederverwendet werden können. Für die automatisierungstechnische Anlage selbst gestaltet sich das Management heute vollkommen anders, besonders wenn klassische Feldbusse zur Kommunikation eingesetzt werden. Dieser Abschnitt soll die wesentlichen Gegebenheiten des Managements in der industriellen Automation mit ihren gegenwärtigen Ausprägungen überblicksweise darstellen. Es ist davon auszugehen, dass mit der fortschreitenden Integration zwischen Automation und übrigen Unternehmensnetzen (vgl. Abschnitt 1.3.2) auch die Trennung zwischen den Management-Domänen in Zukunft weniger stark ausgeprägt sein wird.

Das prägende Merkmal des Managements in der Automation ist die weitgehende Abwesenheit von expliziten Protokollen und Modellen zur Umsetzung von Management-Aufgaben. Gerade in den klassischen Feldbussen sind Funktionen, die üblicherweise zum Management gezählt werden, direkt in das jeweilige industrielle Kommunikationsprotokoll integriert.

Im Hinblick auf die SMFAs, wie sie im OSI-Management definiert sind, werden in der Automation im Wesentlichen das Konfigurations-, Fehler- und Performance-Management durchgeführt. Sicherheits-Management und Abrechnungs-Management existieren in klassischen Automatisierungsnetzen nicht. Grund dafür ist, dass in klassischen Automatisierungssystemen für beides kein Bedarf bestand. Datensicherheit war, solange die Netze vollständig separiert waren, für die allermeisten Anwender kein Thema, das adressiert werden musste, da es aus ihrer Sicht keine Bedrohungsszenarien gab, bzw. diese nicht wahrgenommen wurden. Mit fortschreitender Integration der verschiedenen Datennetze in Unternehmen, ist es nicht auszuschließen, dass in der Automation zukünftig ein Security-Management standardmäßig etabliert wird. Ähnliches gilt für das Abrechnungs-Management von Anlagen, Automatisierungsnetzen oder gar einzelnen Geräten. Angebotene Dienste wurden für den Zweck der Erfüllung der Aufgabe der jeweiligen Anlage angeboten, dies war a priori festgelegt und bedurfte keiner nachträglichen Abrechnung. Vor dem Hintergrund Cloud-Automation und Green-Automation sind zukünftig jedoch durchaus Nutzungsszenarien für ein Abrechnungs-Management denkbar. Auch bei den drei SMFAs, die in der Automation von Bedeutung sind, verhalten sich Ausprägung und Relevanz anders als auf der Unternehmensebene.

Das Netzwerk-Management im Sinne eines Konfigurations-Managements steht in der Automation eindeutig im Vordergrund. Dabei geht es vorwiegend um das Setzen von Werten und Konfigurationen oder das Auslesen derselben. Wie bereits erwähnt, kommen dafür aber keine eigenständigen Management-Protokolle zur Anwendung, welche ausschließlich für das (Konfiguration-)Management vorgehalten werden, sondern Mechanismen wie etwa das azyklische Lesen bzw. Beobachten und Schreiben von Werten und Wertegruppen. Mit den gleichen Methoden kann in der Regel auch auf Prozesswerte zugegriffen werden, die keineswegs im Fokus eines System- oder gar Netzwerk-Managements liegen. Aus der Integration von Management- und Laufzeitaufgaben kann man ableiten, dass prozess- und managementrelevante Informationen nicht durchgehend voneinander getrennt sind. Vor allem

in den klassischen Feldbussen kann man davon ausgehen, dass zum einen kein Informationsmodell besteht, welches auf die Repräsentation von managementrelevanten Informationen ausgerichtet ist und zum anderen Management- und Prozessdaten häufig gemeinsam in einem Informationsraum gehalten werden. Die Aussage zum Nichtvorhandensein von explizitem Management in Feldbussen muss zumindest auf der Begriffsebene je nach Feldbusprotokoll etwas relativiert werden. Feldbusprotokolle bieten durchaus Funktionalitäten, die direkt mit Management betitelt sind (z.B. Feldbus Management (FMA) in PROFIBUS; Netzwerkmanagement (NMT) in CAN, Data Management in CIP). Allerdings fehlen jeweils ein explizites und eigenständiges Informations- und Kommunikationsmodell. Des Weiteren ist eine Anwendung der genannten feldbusspezifischen Management-Methoden auf andere Technologien ausgeschlossen.

Feldbustechnologie-spezifisch sind auch die verbleibenden SMFAs der Automation umgesetzt. Vor allem für das Fehler-Management bieten Feldbusse in der Regel umfangreiche Methoden, die zumindest eine Diagnose ermöglichen. Eine (automatische) Reaktion geht allgemein aber nicht über das Informieren eines menschlichen Operators oder das Fahren in einen betriebssicheren Zustand hinaus. Expertensysteme können eine weiterführende (automatische) Reaktion auf Alarme ermöglichen. Auch die Alarmfunktionalitäten sind direkter Bestandteil des jeweiligen Feldbusprotokolls. Eine Behandlung von Alarmen außerhalb des jeweiligen Automatisierungssystems, beispielsweise im System-Management der übergeordneten IT-Infrastruktur, ist in aller Regel nicht vorgesehen.

Eng verbunden mit Alarmen und Diagnosen ist die letzte SMFA, die in der Automation Beachtung findet: das Performance-Management (vgl. Abschnitt 4.1). Diese Management-Funktion wird aktuell fast ausschließlich implizit durchgeführt. Es existieren zwar vereinzelte Lösungen am Markt, die Aspekte des Performance-Managements mit integrieren, auch hier gilt aber die Begrenztheit auf spezifische Systeme.

Die Integration in ein ganzheitliches Management-Konzept steht nicht im Fokus.

Weitere typische Bestandteile eines umfassenden Managements, wie etwa das entfernte Steuern des Betriebszustandes (Start, Stopp, Betriebsmoduswechsel) sind in Automatisierungssystemen vorhanden, unterliegen aber ebenfalls der Beschränkung bezüglich der Übertragbarkeit zwischen Feldbussen.

Je nach Betrachtungspunkt werden Protokolle zur Basis-Konfiguration von Systemen ebenfalls als Mittel des Netzwerkmanagements angesehen. Jeder Feldbus bietet ein solches Konfigurations-Protokoll. Für die klassischen nicht Ethernet basierten Systeme ist auch dieser Dienst wieder protokollspezifisch. In der Welt der Ethernet-basierten Automatisierungssysteme vertraut man an dieser Stelle auf IT-etablierte Standards. Alle industriell eingesetzten Ethernet-Systeme unterstützen heute mindestens das „Dynamic Host Configuration Protocol (DHCP)" [27] zur grundlegenden Konfiguration von Geräten. Weitere Protokolle für die Basiskonfiguration wie BootP (Bootstrap Protocol) [28] und Discovery and basic Configuration Protocol (nur PROFINET [29] [30]) kommen ebenfalls zum Einsatz, haben jedoch einen geringeren Verbreitungsgrad.

Als einziger relevanter Vertreter der übergreifenden Netzwerk-Management-Protokolle kommt im industriellen Kontext das Simple Network Management Protokoll (SNMP [31], [32], [8], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44]) zum Einsatz. Auf technische Details und die generelle Eignung für das Netzwerk-management von Ethernet-basierten Automatisierungsanlagen wird im Abschnitt 4.2 eingegangen.

Sofern vom Gerätehersteller vorgesehen, kann jedes Ethernet-Gerät mit SNMP ausgerüstet werden. Für bestimmte Kategorien von PROFINET IO-Geräten ist SNMP ein zwingend erforderliches Merkmal, ohne dass zum Beispiel Zertifikate für bestimmte Konformanzklassen nicht erreicht werden können.

SNMP ist nach seiner Spezifikation grundsätzlich nur auf das Management von Ethernet-basierten Geräten ausgerichtet. Im Bereich der industriellen Automation gibt es jedoch Bemühungen, klassische Feldbusse auf SNMP abzubilden [45], [46] und somit nicht-Ethernet-basierte Kommunikationssysteme mit Internettechnologien managebar zu machen. Am Markt existieren heute einige Lösungen, zum Beispiel für die PROFIBUS-SNMP-Integration. Aber auch für die meisten anderen relevanten Feldbusse kann man solche Produkte finden. Ein genereller Erfolg für das Konzept Feldbus-SNMP-Integration lässt sich daraus zwar nicht ableiten, ein Interesse an der Integration von Feldgeräten in ein umfassendes und offenes Netzwerk-Management aber schon. Dies gliedert sich in den in erwähnten anhaltenden Trend zur technologischen und strukturellen Integration zwischen Feld- und Unternehmensebene ein.

Zu den häufigsten gegenwärtig in der Automation eingesetzten Management-Schnittstellen (bezieht sich auch, aber nicht ausschließlich auf Netzwerk-Management) zählt das BUI (Browser User Interface). Ein Gerät ist dabei mit einem integrierten Webserver ausgerüstet und ermöglicht den Zugriff auf managementrelevante Informationen. Bei dieser Art von browserbasiertem Management ist in der Regel davon auszugehen, dass auf einer pro-Gerät-Basis gemanagt wird, was bei einem großen Gerätepool einen erheblichen Aufwand darstellt.

Andere Management-Schnittstellen wie Telnet oder serielle Zugänge kommen auch in der Automation zum Einsatz. Sie stellen allerdings schon heute keinen bedeutenden Anteil dar. Grund dafür sind sicher auch die geringe Benutzerfreundlichkeit, Sicherheitsrisiken und schlechte Wartbarkeit.

Egal ob BUI oder Konsolenzugang, von Einheitlichkeit und Durchgängigkeit kann nicht die Rede sein. Oberflächen, Strukturen, zur Verfügung gestellte Informationen und generell der gebotene Funktionsumfang unterscheiden sich nicht nur zwischen Herstellern, sondern auch zwischen Produktlinien eines Herstellers und häufig sogar zwischen einzelnen Softwareständen eines einzelnen Produktes.

Dennoch finden sich in der Praxis immer wieder einzelne Funktionen, die ausschließlich über diese Schnittstellen genutzt oder konfiguriert werden können.

Netzwerk-Management-Systeme, wie man sie aus dem IT-Umfeld kennt, existieren in der Automation in gewisser Weise. Beschränkt man den Diskursbereich jeweils auf eine Technologie bzw. auf einen Hersteller kann man sagen, dass Werkzeuge zum Engineering und zur Prozessbeobachtung im Wesentlichen auch Funktionen eines NMS mit abdecken.

3.2 Anforderungen an das Netzwerk- und System-Management der industriellen Automation in Gegenwart und Zukunft

In diesem Abschnitt sollen Forderungen und Einschätzungen bezüglich des System-Managements und vor allem an die dazugehörigen technischen Mittel für die zukünftige Anwendung in der Automation eingeführt und detailliert erläutert werden. Die einzelnen Forderungen und Einschätzungen werden, sofern dies sinnvoll erscheint, mit Gewichtungen versehen, um eine unter- oder überdurchschnittliche Relevanz auszudrücken. Sofern nicht explizit gewichtet, sind Aussagen als durchschnittlich relevant zu betrachten.

Die Bewertung erfolgt sowohl anhand technischer Kriterien und Anforderungen, wie auch auf Basis eher als weich zu bezeichnender Faktoren, wie zum Beispiel der allgemeinen Komplexität eines Ansatzes. Auf diese Weise soll eine Matrix eingeführt werden, anhand derer im folgenden Abschnitt Management-Ansätze eingeordnet und bezüglich ihrer Eignung für die Automation bewertet werden. Ergänzend muss noch hinzugefügt werden, dass eine solche Aufstellung von Kriterien stark von der Zieldomäne und dem zu erwartenden Nutzerverhalten abhängt, dies gilt auch für die getroffenen Gewichtungen.

Bezüglich eines System-Managements für die industrielle Automation lassen sich folgende Anforderungen ableiten:

Plattformbindung (Plattformbindung)

Inwiefern ist ein Management-Ansatz an eine spezifische Plattform gebunden? Ist ein Management Ansatz beispielsweise ausschließlich für das Management von PC-System geeignet (Bindung an Desktop- und Serverbetriebssysteme), kann sich die Nutzung im Umfeld von Feldgeräten schwierig oder unmöglich gestalten. Durch den (zunehmenden) Einsatz von verschiedenen Plattformen in der Automation stellt die Plattformunabhängigkeit ein überdurchschnittlich wichtiges Kriterium dar.

Domänendurchdringung und Verfügbarkeit (Durchdringung)

Dieses Kriterium drückt aus, in wie fern eine bestimmte Management-Technologie innerhalb einer Domäne (Automation und Enterprise-IT) bereit steht. Das beinhaltet auf der einen Seite, wie häufig eine Technologie zum Standardrepertoire einer Plattform gehört (z.B. standardmäßig installiert ist), grundsätzlich - wenn auch mit Aufwand - verfügbar ist oder eine Unterstützung wenigstens prinzipiell möglich ist. Andererseits ist auch nicht außer Acht zu lassen, wie stark ein Management-Ansatz in der jeweiligen Domäne tatsächlich eingesetzt wird, sie bereits durchdrungen hat. Während sich die generelle Verfügbarkeit noch recht gut bewerten lässt, sind Aussagen zur Nutzung im besten Fall anhand von Verbreitungen (Verkaufszahlen, Downloadzahlen) bestimmter Produkte möglich.

Eine fundierte Bewertung, sowohl qualitativ wie auch quantitativ, von Verfügbarkeit und Domänendurchdringung ist schwierig. Dennoch ist dieses Kriterium mit Hinblick auf die Integration von Enterprise-IT und Automation ein wesentlicher Aspekt. Für die Akzeptanz eines IT-Ansatzes innerhalb der Automation ist es nicht nur wichtig, dass dieser Ansatz in seiner Ursprungsdomäne eine breite Unterstützung erfährt und etabliert ist, sondern auch, dass er für die Automation verfügbar gemacht werden kann oder unter Umständen bereits in gewissem Umfang verfügbar ist. Neue Ansätze im System- oder Netzwerk-Management brauchen in der Regel

Jahre, bis sie - zunächst einmal unabhängig von der Domäne - eine hinreichende Verbreitung erfahren haben.

Mächtigkeit des Basis-Informationsmodells (Mächtigkeit)

Als einer der wesentlichen Bestandteile des System-Managements wurde bereits das Informationsmodell identifiziert. Neben den generellen Eigenschaften, wie Erweiterbarkeit und Flexibilität, ist es wichtig zu beurteilen, wie aussagekräftig und umfassend das Basis-Informationsmodell ist. Damit ist der Teil eines Informationsmodells gemeint, welcher sozusagen mitgeliefert wird. Diese mitgelieferten Basis-Informationsmodelle können einen Standard bzw. de facto Standard darstellen. In der Regel sind sie dadurch gekennzeichnet, dass sie verbreitet, akzeptiert und frei von anwendungs- und herstellerspezifischen Änderungen sind.

Es stellt sich die Frage, warum dem Basis-Informationsmodell eine so große Bedeutung zuzumessen ist. Grundsätzlich sollte es doch ausreichend sein, ein ausdrucksstarkes Metamodell als Konstruktionsanleitung für das jeweilige Informationsmodell zur Verfügung zu haben. Hat man allerdings ausschließlich das Metamodell zur Verfügung, bieten sich dem Anwender schlicht zu viele Freiheitsgrade, was sich negativ auf die Übertragbarkeit von Modellen auswirkt. Ein Metamodell kann in der Regel nicht erreichen, dass gleiche Sachverhalte auch gleich abgebildet werden, lediglich dass die Abbildung den gleichen Regeln gehorcht und einzelne so modellierte Bestandteile mit der gleichen Methode adressiert werden können. Zu viele Freiheitsgrade schränken die Interoperabilität ein, oder erfordern erneut Abstraktionen. Man kann also annehmen, dass ein stärkeres Basis-Informationsmodell dazu führt, dass – sofern sich Anwender an andere grundlegende Modellierungsregeln halten – Interoperabilität zwischen unabhängig entstandenen Modellerweiterungen einfach zu erlangen ist. Ein starkes Basis-Informationsmodell „erzwingt", dass gleiche Sachverhalte auch gleich dargestellt werden.

Eine quantitative Bewertung ist an dieser Stelle ohne weiteres, etwa über die Anzahl an vordefinierten Elementen in (de facto) Standards, möglich. Auf eine Aussage bezüglich der Qualität kann man so aber kaum schließen. Die qualitative Bewertung des Basis-Informationsmodells ist generell schwierig. Als einzige Variante würde sich ein Marktvergleich anbieten. Bei einem solchen Ansatz müsste man überprüfen, ob für ein zu einem Management-Ansatz gehöriges Informationsmodell in verschieden im Markt eingesetzten Modellausprägungen gleiche Sachverhalte auch (ohne weitere Abstraktion) wirklich gleich dargestellt sind. Die Komplexität dieses Vorgehens ist allerdings so groß, dass eine objektivierte Aussage bezüglich der Qualität eines Basis-Informationsmodells im Rahmen dieser Arbeit ausgeschlossen ist. Letztlich muss die Mächtigkeit eines Modells durch den jeweiligen Anwender beurteilt werden und ist nicht unabhängig von dessen Erfahrung. Die Beurteilung dieses Kriteriums ist im Wesentlichen also subjektiver Natur.

Die Anforderung an die Mächtigkeit des Basis-Informationsmodells wird als überdurchschnittlich wichtig eingeschätzt, da aber bezüglich Qualität keine objektivierte Aussage möglich ist, wird an dieser Stelle neutral gewichtet.

Erweiterbarkeit des zugrunde liegenden Informationsmodells (Erweiterbarkeit)

Dieses Kriterium beschreibt, inwiefern das Basis-Informationsmodell erweitert werden kann. Dabei ist es zunächst einmal unabhängig von der Ausdrucksstärke des Metamodells. Es soll lediglich wiederspiegeln, wie gut oder schlecht sich neue Elemente in das Informationsmodell einarbeiten lassen und inwiefern diese Erweiterungen transferiert werden können. Letzteres meint, inwiefern Erweiterungen z.B. eines Herstellers in eine Informationsmodellausprägung eines anderen übertragen werden können ohne sich zu behindern oder gar auszuschließen. Die Fähigkeit, Relationen und Beziehungen zwischen Bestandteilen des Informationsmodells auszudrücken sind explizit nicht Element dieses Kriteriums.

Struktur und Mächtigkeit der Management-Objekte (Objekte)

Bewertet wird durch dieses Kriterium, wie flexibel und aussagestark die Management-Objekte eines Informationsmodells selbst sind. An dieser Stelle ist Objekt nicht zwingend im Sinne von Objekt-orientierung zu verstehen, vielmehr sind auch einfache Bezeichner-Wert-Paare als Management-Objekte etabliert. Aussagestarke Management-Objekte, die neben einfachen Werten auch eine Verhaltensdefinition, Definitionen für ausführbare Funktionen oder Beschreibungen über ihre Relation mit Nachbarn enthalten, werden besser bewertet als einfache Objekte, die lediglich als Daten-container dienen.

Fähigkeit des Informations-Modells zum Ausdruck von Beziehungen zwischen Modellelementen (Beziehungen)

Dieses Kriterium hat starken Bezug zu den vorangegangenen Kriterien. Es bewertet, inwieweit das (Meta-)Modell explizit Beziehungen zwischen einzelnen Management-Objekten ausdrücken kann. Erlaubt ein Informationsmodell lediglich das Aufreihen von Werten, wird es schlechter bewertet als eines, das Vererbung und Assoziation als Beschreibungsmittel zur Verfügung hat. Da dieses Kriterium für die Ausdrucksstärke des gesamten Informations-modells und somit auch für den jeweiligen Management-Ansatz von Wichtigkeit ist, wird es als überdurchschnittlich relevant gewichtet.

Technologiebindung (Technologiebindung)

Ähnlich der Plattformbindung drückt Technologiebindung aus, inwiefern ein Management-Ansatz an eine bestimmt Technologie gebunden ist. Grundsätzlich ist Technologiebindung in diesem Sinne kein negatives Merkmal, wenn zum Beispiel von einer Bindung an eine Technologie die Rede ist, die selbst als flexibel gilt (z.B. Bindung an bestimmte Markup-Sprachen). Inwiefern eine Technologie-bindung für die Automation zur Herausforderung wird, ist sowohl von der gebundenen Technologie, wie auch vom Teilaspekt, in dem sie eingesetzt wird, abhängig. Damit kann die Bindung an bestimmte Stacks, Protokolle, (Programmier-)Sprachen usw. gemeint sein.

Eine weniger stark ausgeprägte Bindung an Technologien führt zu einer besseren Bewertung.

Flexibilität des Management-Ansatzes allgemein (Flexibilität)

Dieses Kriterium spiegelt die Bewertung der prinzipiellen Anwendbarkeit eines Ansatzes auf verschiedenen Domänen und Management-Probleme wieder. Damit hängt es mit der Flexibilität des Informationsmodells, aber auch mit der Technologie- und Plattformbindung zusammen. Generell ist eine größere Flexibilität besser und wird dementsprechend auch besser bewertet.

Möglichkeit der Integration von bestehenden Installationen/Technologien (Integration)

Ein weiterer wesentlicher Punkt für den Erfolg eines Management-Ansatzes in der Automation ist die Fähigkeit bestehende, also im Feld betriebene bzw. als weit verbreitet angesehene Technologien zu integrieren. Integration bezieht sich dabei nicht auf den Einsatz von Proxies, sondern auf das direkte An- bzw. Einbinden in das eigene lokale Informationsmodell. Aus Anwendersicht muss nur noch mit dem „neuen" Informationsmodell gearbeitet werden, ein Rückbezug, sofern notwendig, kann aber wieder hergestellt werden. Ist die Möglichkeit der Integration anderer Informationshaushalte nicht oder nur indirekt (etwa über Proxies) möglich, bestehen zwei mögliche Wege: (i) paralleler Betrieb von mehreren Ansätzen und (ii) Ersetzen des etablierten Ansatzes. Während (i) heute in vielen Bereichen gängige Praxis ist, wird so die Anzahl der zu wartenden (Management-)Systeme unnötig vergrößert. Die schon vielfach beschriebenen Eigenheiten der Automation machen den vollkommenen Ersatz einer Technologie schwierig. Integrationsfähigkeit ist integraler Bestandteil für den Erfolg einer neuen Technologie. Sind Möglichkeiten zur Integration von externen Informationshaushalten gegeben, wird positiv gewertet. Aufgrund der angeführten Bedeutung für die Automation wird dieses Kriterium als überdurchschnittlich relevant eingeschätzt.

Einheitlichkeit der Nutzerschnittstelle (Einheitlichkeit)

Die Schnittstelle zwischen Nutzer- bzw. Anwendungssoftware und dem eigentlichen Management-System sollte so weit wie möglich den einzigen Berührungspunkt mit dem System-Management bilden. Deshalb ist es notwendig, dass diese Schnittstelle einheitlich ist, auf der anderen Seite aber den vollen Zugang zum Funktionsumfang des Management-Systems ermöglicht. Dies bedeutet letztlich, ein Management-Ansatz muss eine vordefinierte Menge an Funktionen zur Verfügung stellen, mit Hilfe derer alle Aktionen am oder auf Basis des Management-Systems ausgeführt werden können. Diese Menge an Interaktionsmöglichkeiten sollte sich auch zwischen verschiedenen Systemen gleichen Typs nicht unterscheiden. Ist eine Einheitlichkeit gegeben, wird positiv bewertet.

Durchgängigkeit des Ansatzes (Durchgängigkeit)

Dieses Kriterium steht in Verbindung mit Flexibilität des Informationsmodells und wird direkt bedingt durch die Einheitlichkeit des Ansatzes. Durchgängigkeit meint hier, inwiefern sich auf den ersten Blick vollkommen unterschiedliche Management-Aufgaben mit den gleichen Mitteln behandeln lassen. Ist Einheitlichkeit gegeben, kann man davon ausgehen, dass von Seiten der Interaktion mit Clients auch Durchgängigkeit gewährleistet werden kann. Alle „anderen" Schnittstellen, Beschreibungen etc. wären in diesem Fall bereits vereinheitlicht. Darüber hinaus adressiert Durchgängigkeit aber auch, in welchem Umfang grundverschiedene Management-Aufgaben innerhalb des Systems gleich gehandhabt werden. Unterscheidet sich auf der Ebene des Management-Systems etwa der Zugriff auf die Temperaturwerte eines eingebetteten Gerätes vom Zugriff auf gegenwärtige Firmware-Version desselben Systems, ist Durchgängigkeit eingeschränkt. Letztlich hängt die Durchgängigkeit direkt mit dem Vorgehen bei der Instanziierung des Informationsmodells, mit Informationen aus den physischen oder logischen Systemen, zusammen. Sind konkrete Schnittstellen für die Instanziierung und

damit auch eine klare Trennung zwischen Modell und realen Geräten spezifiziert, kann man davon ausgehen, dass das Management-System intern verschiedene Management-Aspekte gleich behandeln kann.

Standardisierung (Standardisierung)

Die industrielle Automation gilt im Allgemeinen als sehr Standardorientiert, eine möglichst fortgeschrittene Standardisierung ist somit auch für System-Managementansätze wünschenswert. Es geht aber nicht nur darum, ob generell standardisiert wurde oder sich ein Ansatz als de facto Standard etabliert hat, sondern vor allem auch darum, wie offen, also für die Mitarbeiter/Mitglieder zugänglich, das jeweilige Gremium ist. Ist ein Ansatz von einer offenen Organisation standardisiert, wird er besser bewertet als einer, der nicht oder nicht von frei zugänglichen Organisationen festgelegt wurde.

Objektauswahl (Objektauswahl)

Das Kriterium Objektauswahl - wiederum nicht zwingend im Sinne von Objektorientierung – bewertet, wie einzelne oder Gruppen von Objekten aus dem instanziierten Informationsmodell ausgewählt bzw. abgefragt werden können. In objektorientierten Informationsmodellen wird man in der Regel frei auswählen können, bzw. spezialisierte (durch Vererbung) Abfrageergebnisse zur Verfügung gestellt bekommen. In Informationsmodellen, die auf Tabellen oder Baumstrukturen basieren, kann iterativ oder blockweise zugegriffen werden. Es ist jedoch nicht zwingend, dass ein semantischer Zusammenhang zwischen den Datenelementen besteht. Bezüglich Wahlfreiheit und Flexibilität ist der freie Zugriff in objektorientierten Informationsmodellen besser zu bewerten als eine iterative Vorgehensweise.

Modellbereitstellung und Modellbedarf (Modellbereitstellung)

Werden von Seiten des Nutzers bzw. Clients Teile oder das vollständige Informationsmodell benötigt, um auf das instanziierte

Modell zugreifen oder dessen Daten interpretieren zu können? Stellt eine Komponente des jeweiligen Management-Ansatzes alle relevanten Teile des Informationsmodells zur Verfügung? Dies sind die beiden Kernfragen, die sich hinter dem Bewertungskriterium verbergen. Die erste Frage ist fast generell mit „Ja" zu beantworten, denn ohne das eigentliche Informationsmodell wird ein Nutzer einzelnen Daten oder ganzen Objekten sicherlich nur wenig Bedeutung beimessen können. Die wichtigere ist aber die zweite Frage: Muss sich der Nutzer (bzw. auch der Hersteller) selbst darum bemühen das Informationsmodell, etwa in Form einer externen Beschreibungsdatei, bereitzustellen oder bietet der jeweilige Management-Ansatz einen Weg, alle benötigten Informationen aus sich selbst heraus zur Verfügung zu stellen? Gerade im Hinblick auf komplexe Informationsmodelle, die eine Vielzahl von spezifischen Erweiterungen enthalten, was gerade in der Automation häufig auftreten kann, ist dies ein überdurchschnittlich relevantes Kriterium. Je weniger Aktivität vom Nutzer erwartet werden muss, desto besser wird bewertet.

Dynamisches Erzeugen von Management-Objekten (Dynamik)

Mit diesem Kriterium wird bewertet, inwiefern während des Betriebs eines Management-Systems neue Management-Objekte angelegt werden können. Liegt ein objektorientiertes Informationsmodell zugrunde, stellt sich diese Frage in der Regel nicht, da neu Instanziierungen von Klassen allgemein während des Betriebs geschehen. In diesem Fall bezieht sich das Erzeugen von Management-Objekten auf das dynamische Anlegen bzw. Ändern neuer Klassen im Informationsmodell. Für Informationsmodelle, denen kein objektorientiertes Metamodell als Grundlage dient, bezieht sich das dynamische Anlegen auf die Erzeugung zusätzlicher Bezeichner-Wert-Paare. Positiv wird bewertet, wenn der Management-Ansatz es direkt ermöglicht, sein eigenes Informationsmodell auf diese Weise zu ändern.

Notifikationen (Notifikation)

Die Leistungsfähigkeit eines Notifikationsmechanismus wird hier beurteilt. Sie hängt davon ab, welche Art von Information eine Notifikation übermittelt. Wird lediglich informiert, dass ein Ereignis vorliegt oder wird das Ereignis selbst (eingebettet in die Nachricht) übertragen. Ist ein Notifikationsmechanismus explizit definiert, wird positiv bewertet, neutral für den Fall, dass Notifikationen zwar möglich sind, sich aber im Bereich von „off-spec" Erweiterungen befinden.

Methoden (Methoden)

Besteht im jeweiligen Ansatz die Möglichkeit mit oder auf Basis von Management-Objekten Methoden auszuführen, die über das Setzen von Werten hinausgehen? Für komplexe Management-Aufgaben und vor allem für die Durchsetzung eines Funktionsmodells ist dies von großer Bedeutung. Auch dieses Kriterium bewertet wiederum, ob es möglich ist Methoden zu definieren und auszuführen, nicht jedoch, welche Komplexität diese im jeweiligen Ansatz erreichen dürfen. Lassen sich Methoden im beschriebenen Sinne definieren und ausführen, wird positiv gewertet. Aufgrund der Bedeutung für die Durchsetzung anspruchsvoller Management-Aufgaben und die gesamte Flexibilität im Funktionsumfang eines Management-Ansatzes wird diesem Kriterium eine überdurchschnittliche Relevanz beigemessen.

Transport (Transport)

Die technische Realisierung der Kommunikation zwischen Management-System und dem bzw. den Konsumenten wird hier als Transport bezeichnet. Eine generelle Bewertung, ob ein Transport gut oder weniger gut im Sinne von verursachter Netzauslastung, Fehleranfälligkeit, Verbindungs- und Routingverhalten etc. ist, wird in dieser Arbeit nicht vorgenommen. Bewertet wird, inwieweit der von einem Management-Ansatz genutzte Transport im Automatisierungsumfeld heute schon eingesetzt wird, oder auf der anderen

Seite vollkommen neu in dieser Landschaft wäre. Je weiter in Richtung Anwendungsschicht (im Sinne des OSI-Schichtenmodells [47]) der eingesetzte Transport auf automationserprobte Protokolle zurückgreift, desto besser wird bewertet. Grund dafür sind administrative Anforderungen an das zugrunde liegenden Netzwerk.

Beispiel: Nutzt ein Management-Ansatz das Hypertext Transfer Protocol (http), wird in der Regel die notwendige Anpassung an Firewall, Quality of Service Konfiguration etc. überschaubar bleiben. Wird auf ein in der gleichen Installation vollkommen unbekanntes Protokoll gesetzt, muss gegebenenfalls aufwendig neu konfiguriert werden.

Die verbleibenden Eigenschaften lassen sich nicht mehr als gefordertes Kriterium bezeichnen und werden deshalb auch nicht gewertet, sie dienen primär einer subjektiven Einordnung.

FCAPS-Abdeckung (FCAPS)

In Abschnitt 4.1 wird FCAPS als das Funktionsmodell eingeführt. Bis auf Ausnahmen setzen Management-Ansätze FCAPS nicht explizit oder nur in bestimmten Teilen explizit um. Mit der Einordnung *FCAPS-Abdeckung* soll ausgedrückt werden, welche der SMFAs der jeweilige Ansatz bedient. Da die Grenzen zwischen den fünf Bereichen teilweise fließend sind, ist eine exakte Einordnung nicht immer möglich.

Konzeptkomplexität allgemein (Konzeptkomplexität)

Konzeptkomplexität bewertet subjektiv aufgrund der praktischen und theoretischen Arbeit mit den verschiedenen Ansätzen. Eingeordnet wird von „sehr hohe Komplexität" bis „sehr niedrige Komplexität".

Haupt-Anwendungsbereich(e) in der Praxis (Anwendungsbereiche)

Dieses Kriterium dient lediglich der Einordnung und stellt keine Wertung dar. Es wird an dieser Stelle versucht, die Bereiche, in denen der jeweilige Management-Ansatz heute eingesetzt wird, abzustecken. Die Einordnung erfolgt in Domänen, etwa Enterprise

IT, Office, Telekommunikation, Automation, Wissenschaft. Diese Einordnung steht nicht zwingend im Zusammenhang mit dem Kriterium Domänendurchdringung.

Einen sehr wesentlichen Anteil der Bewertungskriterien macht offenbar das Informationsmodell, bzw. Kriterien, die diesem direkt zuzuordnen sind, aus. Anforderungen an das Funktionsmodell sind implizit bewertet, Organisations- und Kommunikationsmodell fanden bei der Auswahl der Kriterien wenig Beachtung. Die starke Heterogenität in der Automation führt für Organisations- und Kommunikationsmodell dazu, möglichst große Flexibilität zu fordern. Eine Beurteilung, ob die beiden genannten Modelle im jeweiligen Management-Ansatz aber ausreichend flexibel sind, hängt, deutlich stärker als dies beim Informationsmodell der Fall ist, von den konkreten Technologien ab, die im Zielsystem vorgefunden werden.

Abschließend wird die Tabelle vorgestellt, mit deren Hilfe im folgenden Abschnitt die jeweiligen Management-Technologien bewertet und gegenübergestellt werden. Um unnötig komplexe Bewertungsvorgänge zu vermeiden, werden alle Kriterien entweder mit 0, 1 oder 2 bewertet. Die Bedeutungen sind dabei:

→ 0: Technologie erfüllt das Kriterium nicht oder nicht in relevantem Umfang
→ 1: Technologie erfüllt das Kriterium teilweise oder kann das Kriterium unter gewissen Umständen erfüllen
→ 2: Technologie erfüllt das Kriterium vollständig bzw. in für die Zieldomäne relevantem Umfang (gegenwärtig und in absehbarer Zukunft)

Es wird dabei immer positiv gewertet, was bedeutet, dass auch Kriterien, die anscheinend eine gegenläufige Aussage treffen, in der Reinfolge 0,1,2 bewertet werden. Das Kriterium Plattformbindung z.B. wird mit 2 gewertet, wenn keine Plattformbindung besteht.

Tabelle 1 Bewertungskriterien und ihre Gewichtung

Bezeichnung	Kurzbeschreibung	Gewichtung
Plattformbindung	Plattformbindung	über Durchschnitt
Durchdringung	Domänendurchdringung und Verfügbarkeit	neutral
Mächtigkeit	Mächtigkeit des Basis-Informationsmodells	neutral
Erweiterbarkeit	Erweiterbarkeit des zugrunde liegenden Informationsmodells	neutral
Objekte	Struktur und Mächtigkeit der Management-Objekte	neutral
Beziehungen	Fähigkeit des Informations-Modells zum Ausdruck von Beziehungen zwischen Modellelementen	über Durchschnitt
Technologiebindung	Technologiebindung	neutral
Flexibilität	Flexibilität des Management-Ansatzes allgemein	neutral
Integration	Möglichkeit der Integration von bestehenden Installationen/Technologien	über Durchschnitt
Einheitlichkeit	Einheitlichkeit der Nutzerschnittstelle	neutral
Durchgängigkeit	Durchgängigkeit des Ansatzes	neutral
Standardisierung	Standardisierung	neutral
Objektauswahl	Objektauswahl	neutral
Modellbereitstellung	Modellbereitstellung und Modellbedarf	über Durchschnitt
Dynamik	dynamisches Erzeugen von Management-Objekten	neutral
Notifikationen	Notifikationen	neutral
Methoden	Methoden	über Durchschnitt
Transport	Transport	neutral
FCAPS	FCAPS-Abdeckung	(ohne Einfluss)

Bezeichnung	Kurzbeschreibung	Gewichtung
Konzeptkomplexität	Konzeptkomplexität allgemein	(ohne Einfluss)
Anwendungsbereiche	Haupt-Anwendungsbereich(e) in der Praxis	(ohne Einfluss)

4 Technologien und Ansätze zum Netzwerk-, System- und Dienste-Management

Die Begriffe für diesen Abschnitt scheinen sehr weitläufig gewählt zu sein. In der Tat ist es so, dass der Grenzbereich zwischen den einzelnen Begriffen immer stärker verschwimmt. Was für den einen Betrachter noch Netzwerk-Management sein mag, würde ein zweiter bereits als System-Management bezeichnen und so weiter.

Es sollen in diesem Abschnitt Technologien betrachtet werden, die auf die eine oder andere Weise entweder in der klassischen IT oder in der Automation von Bedeutung sind. Teils haben sie nur noch historische Bedeutung, bilden aber die Grundlage für gegenwärtig eingesetzte Management-Systeme. Eine Ausnahme – WIMA/JAMAP – soll als Vertreter der vorwiegend als akademisch zu bezeichnenden Vertreter von Management-Ansätzen betrachtet werden. Der Fokus der Betrachtungen liegt jedoch klar auf verbreiteten Management-Ansätzen. Wie bereits erwähnt, werden NMS nicht explizit betrachtet, da sie in der Regel ein oder mehrere Management-Ansätze integrieren, jedoch keinen eigenen unabhängigen und allgemein akzeptierten Ansatz einführen.

Neben den in dieser Arbeit zu betrachtenden Ansätzen existiert noch eine Reihe von sehr mächtigen Standards mit starkem oder sogar ausschließlichem Telekommunikationsbezug. Darunter befinden sich *Intelligent Network (IN), Telecommunication Information Networking*

Architecture (TINA) und nicht zuletzt *Frameworkx (vormals New Generation Operations Systems and Software)*. Im Fokus liegt bei diesen Ansätzen nicht mehr nur das System-Management, sondern auch ein umfassendes Business Process Management. In [48] wird ein kurzer Überblick über die genannten Ansätze gegeben. Im Rahmen dieser Arbeit werden *IN, TINA* und *Frameworkx* jedoch nicht näher betrachtet.

Warum Netzwerk-, respektive System-Management notwendig ist und nach wie vor an Bedeutung gewinnt, wurde bereits besprochen. Ein wesentlicher Punkt wurde bislang jedoch außen vor gelassen: Allen Standard-basierten Management-Ansätzen ist gemein, dass letztlich auf die Umsetzung und Implementierung des Standards durch Hersteller vertraut werden muss. Diese Erkenntnis ist keineswegs neu und wird schon in [18] und [19] beschrieben. Ein großer Teil der Möglichkeiten, die umfassende Management-Ansätze bieten, lässt sich erst effektiv entfalten und nutzen, wenn sie direkt in den jeweiligen Geräten implementiert sind. Handelt es sich beim jeweiligen Managed Object um Software, ist es notwendig, dass der Hersteller fest definierte Schnittstellen zur Verfügung stellt oder auch in diesem Fall Management-Standards direkt selbst implementiert. Das Out-Of-Band-Management – also das Management von außen ohne direkte Unterstützung im Managed Object, meistens über vorhandene Schnittstellen - ist häufig zwar immer noch eine Möglichkeit, kann aber schnell neue Probleme schaffen. Zu diesen Problemen zählen Belastungen des Kommunikationsnetzes, kritisches Zeitverhalten bei statistischen Werten (z.B. Zeitstempel), Unschärfe (z.B. Paketzähler) und letztlich Einschränkungen, die durch die genutzten Schnittstellen auferlegt sind. Durch die Forderung nach Herstellerunterstützung wird klar, dass im späteren Verlauf dieser Arbeit umgesetzte Anwendungsfälle in weiten Teilen auf die Nutzung des Out-Of-Band-Ansatzes angewiesen sind. Unter Beachtung der genannten Einschränkungen stellt dies jedoch für prototypische Umsetzungen kein Manko dar, da sie von Herstellern inhaltlich schnell übernommen werden könnten.

Es sei noch einmal darauf hingewiesen, dass hier ausschließlich technische Aspekte des Managements betrachtet werden. Die Themenbereiche IT Service-Management (ISO 20000) und ITIL, sowie damit in Verbindung stehende – nicht technische – Konzepte, finden keine Beachtung. Damit in Zusammenhang stehen auch Planung und Organisation [20] von Netzwerken, die zwar wichtige Aspekte eines Netzwerkes sind, aber in der Regel durchgeführt werden, bevor das eigentliche Netzwerk- bzw. System-Management umgesetzt wird. In der Automation wird dies während des Anlagen-Engineerings durchgeführt.

In den vorangegangenen Absätzen wurde es bewusst vermieden von spezifischen Management-Technologien zu sprechen, um keine inhaltlichen Vorgriffe erforderlich zu machen. Dies wird nun nachgeholt. Es wird eine Reihe von Ansätzen verglichen und ihre Eignung für den zukünftigen Einsatz in der Automation bewertet. Dafür werden die in Abschnitt 3.2 eingeführten Kriterien herangezogen.

Zunächst wird jede der betrachteten Technologien kurz eingeführt und versucht, ihr gegenwärtiges Einsatz- oder Zielanwendungsgebiet abzustecken. Anschließend werden die technischen und organisatorischen Aspekte zusammengetragen, die abschließend benötigt werden, um eine Bewertung nach den aufgestellten Kriterien durchführen zu können.

4.1 OSI-Netzwerkmanagement

Das OSI-System-Management (Management framework for Open Systems Interconnection) ist wohl der am meisten zitierte Standard zum Thema Netzwerk- und System-Management überhaupt. Eigentlich handelt es sich dabei nicht um einen einzelnen Standard, sondern um eine Sammlung von Standards. Abhängig davon, welche Teile man als direkt zugehörig zum OSI-System-Management betrachtet sind dies zwischen ca. 30 bis 50 Unterstandards, die teilweise in mehrere Dokumente gegliedert sind. Die Normung erfolgt dabei in ISO-Standards. Parallel dazu sind alle Inhalte auch als ITU (vormals CCITT) Empfehlungen erhältlich. Aus Gründen der Zugänglichkeit werden in

diesem Abschnitt (und auch in allen anderen Teilen der Arbeit, in denen das OSI-System-Management erwähnt wird) inhaltlich ausschließlich ITU-Empfehlungen referenziert. Unter anderem in [19] ist eine Gegenüberstellung zwischen ISO- und ITU-Dokumenten zum Thema zu finden. Da keine Mehrdeutigkeit zu fürchten ist, wird im verbleibenden Teil der Arbeit der Begriff OSI-Management an Stelle von OSI-System-Management verwendet.

Ende der 1980er Jahre waren Unternehmens- und Telekommunikationsnetze in ihrer Komplexität derartig gewachsen, dass die bis zu dieser Zeit vorherrschenden spezifischen Management-Ansätze nicht mehr ausreichten. Hinzu kamen steigende Anforderungen an die Interoperabilität zwischen eingesetzten Komponenten verschiedener Hersteller; eine Entwicklung, sehr ähnlich zu der, wie sie seit einiger Zeit in den großen Kommunikationsinfrastrukturen der industriellen Automation anzutreffen ist. Als Resultat dieses Bedarfes wurde von der International Standard Organization (ISO) mit den Arbeiten am OSI-Management begonnen und 1989 erste Dokumente veröffentlicht. Bis zur Mitte der 1990er wurden sehr aktiv viele weitere Unterstandards zum OSI-Management veröffentlicht. Das OSI-Management gilt auch heute noch als der umfassendste Standard in diesem Bereich.

Die nahezu allumfassende Mächtigkeit der Konzepte im OSI-Management führte jedoch dazu, dass nie eine breite Marktdurchdringung erreicht wurde. Heute hat das OSI-Management im Unternehmensbereich und auch in der Automation keine Bedeutung. Es wurde von einfacheren Ansätzen wie SNMP (vgl. 4.2) verdrängt. Einzig im Bereich der Telekommunikation hat das OSI-Management, genauer das auf OSI basierende Telecommunications Management Network (TMN) [49], heute noch einige Bedeutung. Neben der angesprochenen Komplexität der einzelnen Komponenten des OSI-Managements kommt hinzu, dass, um das OSI-Management implementieren und umsetzen zu können, quasi zwingend ein vollständiger OSI-Stack vorausgesetzt ist. Das wundert nicht, denn als Zieldomäne für das OSI-Management gelten Netzwerke, bzw. alle Entitäten in einem Netzwerk, die einen

vollständigen OSI-Stack [47] bieten. Solche vollständigen Stacks sind aber in der Praxis kaum verbreitet, so dass sie erst aufwendig umgesetzt werden müssten – ein zu großer Aufwand für die meisten Nutzer und der wesentliche Grund, warum der Erfolg des OSI-Managements ausblieb.

Dennoch ist es absolut notwendig das OSI-Management im Rahmen dieser Arbeit zu betrachten. Nicht nur aus dem Grund der Vollständigkeit des Ansatzes heraus, sondern vielmehr, weil in diesem Rahmenstandard Aspekte und Grundlagen definiert sind, die heute in den meisten praktisch eingesetzten Netzwerk- und System-Management-Ansätzen nach wie vor gelten. Das OSI-Management beschreibt in [22] die bereits angesprochenen SMFAs, die heute, wenn in einigen Anwendungsdomänen auch erweitert, nach wie vor gelten und breit akzeptiert sind. Das System-Management-Model ist ebenfalls bereits in [22] beschrieben, es enthält Aspekte zu Informations-, Kommunikations-, Organisations- und Funktionsmodellen, die in der Einleitung zu Abschnitt 3 bereits eine Rolle gespielt haben.

Das OSI-Management war in vielerlei Hinsicht seiner Zeit voraus, es setzte beispielsweise von Anfang an auf eine objektorientierte Beschreibung von gemanagten Entitäten und lieferte somit die Grundlage für eine spätere Erweiterung.

Im folgenden Abschnitt 4.1.1 werden relevante Aspekte aufgrund ihrer Bedeutung für diese Arbeit vertieft. Dies kann aufgrund des Umfangs ausschließlich für die in dieser Arbeit relevanten Aspekte und Bewertungskriterien geschehen, detaillierte Beschreibungen sind entweder den entsprechenden Normen oder [19], [18] zu entnehmen. Für das OSI-Management bietet es sich an, dies anhand der vier Aspekte bzw. Modelle des System-Management-Modells zu gliedern.

4.1.1 Technologische Einordnung

Aus technologischer Sicht ist im OSI-Management vor allem ein Kommunikationsrahmenwerk definiert. Die Beschreibung wie Informationen darzustellen sind [50], wie sie zu handhaben sind bzw. wie auf

ihnen operiert werden kann [51] und wie sie letztlich kommuniziert werden [52] nimmt einen wesentlichen Teil am Standard ein. Funktionale Aspekte ([52] grundsätzlich, ITU-T X.730-751 [53] [54] einzelne Management Funktionen) werden separat adressiert, organisatorische Aspekte [51], [22] eher implizit behandelt.

Zunächst werden die organisatorischen Aspekte des OSI-Managements so knapp wie möglich erläutert. Am Ende des Abschnittes 3 wurde bereits darauf hingewiesen, dass die organisatorischen Aspekte (*Organisationsmodell*) für diese Arbeit eine untergeordnete Rolle spielen, für den Gesamteindruck ist ein Überblick über die Organisationsstrukturen aber notwendig.

In Abbildung 7 ist ganz grundsätzlich die Struktur eines Systems im Sinne des OSI-Managements dargestellt. Manager und Agent sind dabei Rollen, die der so genannte „System-Management-Application-Process" (SMAP) [22] annehmen kann. SMAP stellt sich dabei als ein Prozess dar, der auf dem jeweiligen zu managenden Gerät umgesetzt ist. Im Rahmen des Managements hat SMAP die Aufgabe Operationen auszuführen, den Zugriff auf Parameter des kontrollierten Gerätes zu gewähren und sich mit SMAPs auf anderen Geräten zu koordinieren. Für den eigentlichen Austausch von Management-Informationen zwischen Geräten ist der so genannte „System-Management-Application-Entity" (SMAE) verantwortlich, welcher seinerseits das „Common-Management-Information-Protocol" (CMIP) nutzt. Auf CMIP wird im Rahmen des Kommunikationsmodells vertiefend eingegangen. Der verbleibende wesentliche Teil in der Organisations-Struktur des OSI-Managements ist das N-Layer-Management. Das Layer Management stellt Netzwerk-Management-Funktionen speziell für jedes der sieben OSI Layer zur Verfügung. Neben den relevanten Standards [52] und [22] bieten [18] und [19] sehr viel detailliertere Beschreibungen der OSI-Organisationsstrukturen, die über die in dieser Arbeit benötigte Detailtiefe hinausgehen. In [55] werden weitere Erläuterungen vorgenommen und OSI-Management als Basis für TMN betrachtet. Die Rollen Manager-Agent sowie ihre Verteilung und Wechselbeziehungen werden ebenfalls in [55] behandelt.

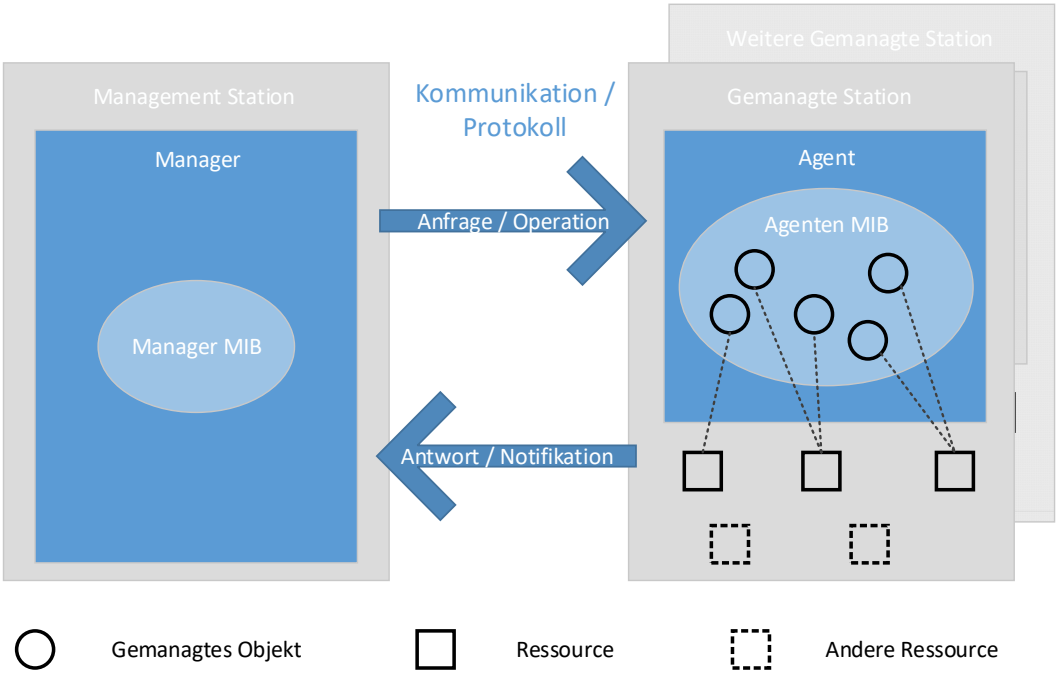

Abbildung 7 Grundsätzliches OSI-Management

Das eigentliche Management erfolgt dabei auf abstrakten Objekten, den Managed Objects (MO), diese bilden eine Repräsentation der realen Ressource in der jeweiligen gemanagten Station. Es sind dabei nur jene Aspekte vom Management betroffen, deren Eigenschaften auf ein MO übertragen wurden. Ein MO kann dabei mehrere reale Ressourcen repräsentieren, es können aber auch Teilaspekte einer realen Ressource auf mehr als ein MO abgebildet sein. Das Nichtvorhandensein von MOs trifft keine Aussage über Ressourcen, vereinfacht ausgedrückt: Es müssen nicht alle Ressourcen auch eine MO-Repräsentation besitzen. Die Gesamtheit aller Management-Informationen wird als Management Information Base (MIB) bezeichnet, sie spiegelt das Informationsmodell wieder. Vom Standpunkt der organisatorischen Aspekte des OSI-Managements ist es so, dass alle aufgezählten Einzelteile zwingend umgesetzt sein müssen (SMAP, SMAE, Layer Management, und MIB). Das gilt für alle Entitäten des Systems/Netzwerkes, die in das Management einbezogen werden sollen. OSI-Management kann per definitionem nur für vollständig OSI-konforme Geräte umgesetzt werden!

Nach den organisatorischen Strukturen sollen nun kurz die wesentlichen Begriffe und Funktionen des *Kommunikations-Modells* erläutert werden, denn auch hier finden sich Parallelen zu anderen Management-Ansätzen.

Das Kommunikationsmodell bzw. die Kommunikationsaspekte bestehen im OSI-Management, wie die meisten anderen Komponenten auch, aus mehreren Aspekten und dazugehörigen Standards. Wesentlich sind hier das Common Management Informations Service (CMIS) [51] und das Common Management Information Protocol (CIMP) [56]. CMIS definiert dabei die im Rahmen OSI-Managements angebotenen Dienste, während CMIP die Definition der übertragenen PDUs vornimmt, mit Hilfe welcher die eigentlichen Management-Informationen übertragen werden. CMIP wird hier keine weitere Rolle spielen, da es nicht unwesentlich von der jeweiligen Ausprägung des OSI-Stacks abhängt. Entwickler sind/waren lediglich daran gehalten, dem so genannten PICS (Protocol Implementation Conformance Statement) Folge zu leisten, die in [57] definiert sind. Deutlich interessanter für diese Arbeit ist CMIS.

Durch CMIS werden in erster Linie s.g. Management Operations festgelegt, mit Hilfe derer auf Basis der MO gearbeitet wird. Diese Operationen sind aus Sicht eines Anwenders bzw. eines Managers die einzigen Operationen, die in einem OSI-System durchgeführt werden können. Durch die Strukturierung in Objekten bzw. den drei OSI Bäumen (siehe Abschnitt zu Informationsmodell im OSI-Management) kann mit dieser (auf den ersten Blick beschränkten) Anzahl an Operationen der notwendige Mächtigkeitsgrad erreicht werden, um auch sehr komplexe Systeme zu managen.

Tabelle 2 CMIS Dienste

Operation/ Dienst	Bedeutung
M-GET	Abrufen von Management-Informationen
M-SET	Modifikation von Management-Informationen
M-ACTION	Ausführen einer vordefinierten Aktion

M-CREATE	Erzeugen einer MO-Instanz
M-DELETE	Entfernen einer MO-Instanz
M-CANCEL-GET	Abbruch einer vorher eingeleiteten M-GET Operation
M-EVENT-REPORT	Meldung eines Ereignisses im MO

Neben den in Tabelle 2 beschriebenen Operationen gibt es im OSI-Management noch Mechanismen, um den Objektzugriff zu optimieren. Vor allem vor dem Hintergrund der Systeme, für die das OSI-Management entworfen wurde, sind diese relevant.

Scoping: Ermöglicht das Selektieren von Sub-Bäumen des gesamten Informationsbaumes, das selektierte Element wird sozusagen zum temporären Wurzelelement.

Filtering: Filterung auf Basis von Eigenschaften/Werten, der zu einem MO gehörenden Attribute. Die Verknüpfung mehrerer Filter über einfache logische Ausdrücke ist möglich.

Synchronization: Dient zum Auflösen von Race-Condition-ähnlichen Zuständen, die aus der Mehrfachselektion mittels Scoping und Filtering entstehen können.

Für tiefere Einblicke in CMIS/CMIP sei an dieser Stelle wieder auf [18], [19] und natürlich auf die zugehörigen Standards [51], [56] verwiesen.

Bevor die technische Einführung in das OSI-Management mit dem für diese Arbeit wichtigsten Teil - dem Informationsmodell - abgeschlossen werden kann ist es notwendig auf das schon mehrfach erwähnte *Funktionsmodell*, wenigstens auf einer abstrakten Ebene, einzugehen. Bei der Definition des OSI-Managements wurde sich zu Beginn die Frage gestellt, welche Aufgaben mit dem Ansatz bearbeitet werden sollen und welche Anforderungen daraus ableitbar sind. Aus diesen Überlegungen sind die SMFAs entstanden, die selbst keine detaillierte oder gar technische Beschreibung von Management-Funktionen sind, sondern vielmehr die Aufteilung von Verantwortlichkeiten beschreiben. Die

SMFAs werden üblicherweise unter dem Akronym FCAPS zusammengefasst, wobei sich der Begriff [52] zusammensetzt aus:

***F**ault:* Fault-Management beschäftigt sich mit den Problemen, die im Kommunikationsnetz vorliegen. Aufgabe des Fault-Management ist es dabei, das jeweilige Problem festzustellen, zu isolieren und letztlich zu beheben.

***C**onfiguration:* Configuration-Management identifiziert, kontrolliert, sammelt und gewährt Zugriff zu Daten im Zielsystem.

***A**ccounting:* Accounting-Management befasst sich mit der Abrechnung (primär von Kosten), die durch die Nutzung des Systems oder angebotener Dienste entstehen.

***P**erformance:* Performance-Management stellt dem Manager Mittel zu Verfügung, um die gegenwärtige Leistungsfähigkeit des Systems zu bewerten.

***S**ecurity:* Security-Management beschäftigt sich mit dem Anwenden von Security Policies und mit dem Management all der Dienste, die Sicherheitsdienstleistungen (etwa Zugangskontrolle) anbieten.

[19] weist darauf hin, dass in [52] FCAPS nicht selbst standardisiert sind, sondern lediglich beschrieben werden. Das Umsetzen der eigentlichen – technischen – Management-Funktionalitäten erfolgt mit Hilfe der System Management Functions (SMF), die ihrerseits jeweils in eigenen Standards (ITU-T X.730-751 [53] [54]) beschrieben sind. [19] weist weiterhin darauf hin, dass eine SMF durchaus Aufgaben und Anforderungen aus mehr als einer der fünf SMFAs adressieren kann.

Auch heute gilt FCAPS noch als Orientierung, wenn es um funktionale Aspekte im Diskursbereich von Netzwerk- und System-Management geht. Es gilt Domänenübergreifend als de facto Standard. [20] erweitert FCAPS noch um Asset- und Support-Management. Obwohl die Argumentationen grundsätzlich schlüssig erscheinen, lässt der Autor eine Anwendung seiner Erweiterungen sowie die Einordnung bestehender Ansätze offen. Für die vorliegende Arbeit werden diese Erweiterungen deshalb nicht in Betracht gezogen.

Der zweifellos wichtigstes Teil jedes Management-Systems sind die zugehörigen Informationen. Diese Informationen liegen in der Regel in einer strukturierten Form vor, welche im Rahmen des OSI-Managements als Management-Information-Base (MIB) bezeichnet wird. Damit die Struktur einer MIB vergleichbar zwischen OSI-Systemen bleibt existiert für alle Aspekte, die zu einer MIB gehören, ein Rahmenwerk, das bei der Definition hilft. Dieses Framework wird als *Structure of Management Information* (SMI) bezeichnet.

Ähnlich wie mit dem Funktionsmodell des OSI-Managements verhält es sich auch mit dem *Informationsmodell*. Die im OSI-Management getroffenen Festlegungen sind auch heute noch Vorbild für Informationsmodelle in Management-Systemen. Auch das OSI-Informationsmodell besteht wieder aus einer Reihe von Standards, die zusammen die SMI bilden. Grundsätzliche Festlegungen sind in [58] getroffen, dieser Standard wird auch als *Management Information Model* (MIM) bezeichnet. Managed Objects selbst sind bereits in [52] beschrieben. Die Definition wesentlicher, etwa durch SFMs genutzter, MO-Klassen, Name-Bindings, Packages und (generischer) Attribute wird in [59] vorgenommen – auch als *Definition of Management Information* (DMI) bekannt. DMI kann als Basisinformationsmodell im Sinne von Abschnitt 3.2 angesehen werden. Den dritten wesentlichen Teil stellt [50] dar, in dem *Guideline for Definition of Managed Objects* (GDMO) beschrieben ist. Letztlich handelt es sich dabei um eine Reihe von *Best-Practise*-Anweisungen und Templates für die Definition von MOs. Zusammen mit dem *General Relationship Model* (GRM) [60] bildet GDMO sozusagen das Metamodell für die Bildung von MO-Klassen und den Ausdruck von Relationen zwischen ihnen.

Wesentlicher Bestandteil jeder MIB sind die Managed Objects. Im Falle des OSI-Managements sind MOs durch eine Reihe von Eigenschaften geprägt. Im Rahmen des Standards sind dies:

- Attribute, die ein MO besitzt,
- Operationen, die auf ihm ausgeführt werden können,
- Notifikationen, die es aussenden kann und schließlich
- die Beziehung zu anderen MOs.

Daneben existieren noch weitere Eigenschaften, die sich implizit ergeben, etwa Aktionen, die durch das MO an der realen Ressource ausgeführt werden, Zugriffsberechtigungen auf das MO, Identifikation des MO etc.

Da es sich beim OSI-Informationsmodell um einen vollständig objektorientierten Ansatz (Klassen, Vererbung etc.) handelt, sind Erweiterbarkeit und Spezialisierung durch Vererbung direkt gegeben. OSI organisiert die MOs dabei in den folgenden drei Bäumen:

- *Der ISO-Registrierungsbaum (auch Namensbaum):* Ist die global eindeutige Identifikation einer MO-Klasse. Der Registrierungsbaum stellt dadurch auch Templates für zukünftige Erweiterungen bereit. Einträge im Registrierungsbaum enthalten den Namen der MO Klasse (vorgegeben durch ihre Position im Baum), Attributdefinitionen für jedes MO, ausführbare Operationen und emittierbare Notifikationen.
- *Der Vererbungsbaum:* Stellt die Vererbungsstruktur von MO-Klassen, im klassischen objektorientierten Sinn, dar.
- Der *Beinhaltungsbaum*: Stellt dar welche MOs logische eine Untergruppe zu einem übergeordneten MO bilden, also welche MOs in einem MO enthalten sein können.

Nachdem die wesentlichen Aspekte des OSI-Managements und vor allem der zugehörigen Modelle nun kurz eingeführt wurden, wird im folgenden Abschnitt die Bewertung bzw. Einordnung dieser Technologie anhand der aufgestellten Kriterien vorgenommen.

4.1.2 Bewertung

Die Bewertung wird anhand der in Abschnitt 3.2 aufgestellten Tabelle durchgeführt und hat somit direkten Bezug zu Herausforderungen (Abschnitt 2.5), wie man sie heute im System-Management in der Automation wiederfindet. Die jeweils getroffene Bewertung wird kurz begründet, um den Bezug zu den im vorangegangenen Abschnitt ausgeführten technologischen Eigenheiten zu verdeutlichen.

Tabelle 3 Bewertung OSI-Management

Kriterium	Begründung	Bewertung
Plattformbindung	OSI-Management ist an keine spezielle Plattform gebunden, zwingend ist nur ein OSI-konformer Stack, der aber grundsätzlich auf jeder Plattform umgesetzt werden kann. Darin liegt aber die Herausforderung. Eine Einführung auf nicht OSI-konformen Systemen ist kurzfristig nahezu ausgeschlossen.	1 ↑
Durchdringung	OSI-Management hat weder in Automation noch in der IT eine relevante Verbreitung, eine Durchdringung kann somit nicht vorhanden sein.	0
Mächtigkeit	OSI-Management bietet zwar eine Reihe von Basisklassen, diese decken aber nur vergleichsweise kleine Teile allgemeiner Systemaspekte ab. Wesentlich sind hier die Definitionen, die vorwiegend in [59] getroffenen sind. Es ist zwar generell eine große Anzahl von Klassendefinitionen zugänglich, diese werden aber eben nur zu einem kleinen Teil von der standardisierenden Organisation selbst gepflegt.	1
Erweiterbarkeit	Das Informationsmodell lässt sich auf direktem Wege erweitern. Aus Sicht des OSI-Informationsmodells ist es kein Problem etwa neue Technologien abzubilden. Das Metamodell unterliegt grundsätzlich	2

Kriterium	Begründung	Bewertung
	keinen Beschränkungen. Wichtiger für diesen Punkt ist aber das Konzept der Objektorientierung.	
Objekte	Nach der Definition in 3.2 erfüllen MOs im OSI-Management dieses Kriterium zweifellos vollständig. Der vorangegangene Abschnitt beschreibt die Eigenschaften und Fähigkeiten von MOs.	2
Beziehungen	Mit GRM bietet OSI vollständige Unterstützung für Beziehungen zwischen MOs.	2 ↑
Technologiebindung	Hier kommt ganz klar zur Geltung, dass OSI-Management auch zwingend einen vollständigen OSI-Stack benötigt.	0
Flexibilität	Das Informationsmodell ist wie beschrieben mächtig, dem gegenüber steht jedoch die recht starke Technologiebindung, so dass Flexibilität zwar grundsätzlich gegeben ist, sich praktisch aber nur bedingt entfalten kann.	1
Integration	Das Einbinden von bestehenden Technologien ist im OSI-Management nicht im Fokus. OSI-Management lässt es in allen Standards offen wie MOs an die realen Entitäten gebunden werden, so dass der Integration von bestehenden Schnittstellen – vom Problem des vollständigen Stacks einmal abgesehen – grundsätzlich nichts im Wege steht. Allerdings wird implizit davon ausgegangen, das OSI-Management für alle relevanten Komponenten Bottom-Up umgesetzt wird, wodurch sich die Frage nach Integration selten stellt. Legacyintegration wird in den Standards nicht adressiert.	1 ↑

Kriterium	Begründung	Bewertung
Einheitlichkeit	Im Wesentlichen können im OSI-Management, mit Hilfe der sieben CMIS-Dienste, alle Operationen an MOs durchgeführt werden die notwendig sind.	2
Durchgängigkeit	Grundsätzlich kann OSI-Management auf alles angewendet werden (Informationsmodell ist erweiterbar und flexibel), auch hier wirkt sich jedoch die Stack-Bindung negativ aus (Toaster haben keinen OSI-Stack [19] [61]).	1
Standardisierung	Das OSI-Management ist in jedem Bereich, den es adressiert stark standardisiert, sowohl grundsätzlich, wie auch auf Ebene von Anweisungen zur Umsetzung in Implementierungen.	2
Objektauswahl	Die Objektauswahl ist vollständig frei und zusätzlich durch Scoping und Filtering optimierbar.	2
Modellbereitstellung	Durch die Grundsätze der Objektorientierung und durch Allomorphie ist eine teilweise Kenntnis des Informationsmodells oft ausreichend, allerdings kann dies nicht zugesichert werden. Es wird im Standard kein Dienst definiert, der es direkt erlaubt, das aktive Informationsmodell aus dem Agent zu laden (etwa ein M-GetClass o.ä.).	1 ↑
Dynamik	Sofern MO-Klassen existieren, können MOs jederzeit über die entsprechenden CMIS-Dienste neu angelegt/instanziiert werden.	2
Notifikationen	Das OSI-Management bietet komplexe Notifikationen, die vollständige MOs beinhalten können.	2
Methoden	Methoden von beliebiger Mächtigkeit können auf MOs	2 ↑

Kriterium	Begründung	Bewertung
	definiert und ausgeführt werden.	
Transport	Eine Bewertung ist schwierig. Management-Informationen werden im OSI-Management generell auf der Anwendungsschicht übertragen (zwischen Management-System und Nutzer), was im Sinne der Definition in 3.2 positiv ist. Allerdings ist das zweite dort aufgeführte Kriterium kaum erfüllt, denn bislang hat OSI-Management in der Automation generell keine Bedeutung, also auch die eingesetzten Protokolle nicht.	1
FCAPS	Das OSI-Management deckt FCAPS vollständig ab.	(ohne Einfluss)
Konzeptkomplexität	Die Konzeptkomplexität wird generell als sehr hoch angesehen, was eben auch am zwingenden Vorhandensein eines vollständigen OSI-Stacks festgemacht werden kann. Ein solcher Stack müsste jeweils implementiert werden.	(ohne Einfluss)
Anwendungsbereiche	Wird im Wesentlichen nur noch im Rahmen von TNM angewendet. TNM selbst verliert aber ebenfalls an Bedeutung [61].	(ohne Einfluss)

Im Mittel ergibt sich für die gewerteten Kriterien ein Wert von 1,4 für alle 18 Kriterien, betrachtet man die 5 Kriterien mit überdurchschnittlicher Relevanz gesondert ergibt sich eine mittlere Bewertung von 1,4. Die historische Bedeutung einmal außen vor gelassen, bedeutet dies, das OSI-Management erfüllt die aufgestellten Kriterien in den wesentlichen Belangen. Wie sich die Bewertung gegen andere Management-Ansätze behauptet, wird am Ende dieses Abschnitts vergleichend dargestellt.

4.2 Simple Network Management Protokoll

Das Simple Network Management Protokoll (SNMP), auch als Internet-Management bezeichnet, ist heute zweifellos das bekannteste Mittel zum Netzwerk-Management. Kein anderer Ansatz ist in so vielen Geräten aller Kategorien verbreitet, kein anderer Ansatz erfährt eine ähnliche Unterstützung in NMS. SNMP ist im Unternehmensbereich genauso etabliert wie in der Automation. Vor allen anderen Gründen ist seine einfache Handhabe dafür verantwortlich.

SNMP ist eine Technologie der IETF (Internet Engineering Task Force). Ursprünglich war SNMP nur vorgeschlagen um als Zwischenlösung zu fungieren bis CMIP bzw. CMOT [62] (CMIP over TCP) fertig gestellt waren. Durch die gegenüber dem OSI-Management geringen Anforderungen sicherte sich SNMP schnell einen Markt und avancierte zum wichtigsten Standard im Netzwerk-Management. Die einstige Behelfslösung ist heute *der Standard* zum Netzwerk-Management.

SNMP wurde in der Vergangenheit mehrfach erweitert, um den während des Einsatzes aufgedeckten Engpässen begegnen zu können. Dazu zählen bekannte und auch praktisch breit eingesetzte Erweiterungen wie SNMPv2c und SNMPv3, aber auch eine Reihe von Erweiterungen, die es in der Praxis nie geschafft haben sich zu etablieren oder eine wirklich als flächendeckend zu bezeichnende Verbreitung zu erreichen, darunter (Secure)SNMP(v2(p,u)) ([63], [64], [65], [66], [67], [68], [69], [70], [71]), Script MIBs [72] und SMP ([19] nie standardisiert, Teile von SMP sind später in SNMPv2 überführt worden). Die umfangreichsten Erweiterungen, die SNMPv2 (außer „2c“) und SNMPv3 vornehmen, betreffen den Bereich der Sicherheit von SNMP selbst. Während v1 und v2c de facto keine Möglichkeit bieten, die Management-Kommunikation zu verschlüsseln und eine Zugriffkontrolle auf bestimmte Management-Aspekte zu realisieren, wird dies von den Versionen 2 (nicht „2c“) und 3 nachgeholt. Ergänzend sei erwähnt, dass der Zusammenhang mit einem Security-Management (vgl. FCAP**S**) - wenn überhaupt - nur indirekt vorhanden ist. Allerdings ist es so, dass eben diese neu eingeführten Security-Features – vor allem in Version 2 - aufgrund der

gestiegenen Komplexität von SNMP oft in der Kritik standen. SNMPv2 ist quasi nicht mehr praktisch im Einsatz, alle wesentlichen Erweiterungen, die abseits der Security-Überlegungen standen, wurden in SNMPv2c übernommen, welches heute noch im Einsatz ist.

Für den Rest dieses Abschnitts und auch für alle weiteren Teile dieser Arbeit werden SNMP(v1), SNMPv2c und SNMPv3 nicht gesondert betrachtet. Alle Änderungen, die die jeweilige Version - abseits der Sicherheitserweiterungen - an den vier relevanten Management-Modellen vornehmen, sind nicht von einem Umfang, der die grundlegenden Konzepte vollkommen verwirft. Deshalb werden, sofern notwendig, lediglich kurze Hinweise gegeben wenn eine relevante Eigenschaft spezifisch zu einer SNMP-Version zuordenbar sein sollte.

Es muss noch einmal herausgestellt werden, dass SNMP aus praktischer Sicht ausschließlich Netzwerk-Management und kein System-Management realisiert (folgt man strikt der OSI-Definition, ist SNMP ein Schicht-7-Protokoll und damit theoretisch System-Management). Dennoch gab es immer wieder Versuche auch in diese Richtung Erweiterungen vorzunehmen. Script MIBs [72] sind eine schon erwähnte Möglichkeit, SNMP um SMF-ähnliche Management-Funktionen zu erweitern. System-Management „by side effects" ist eine andere Vorgehensweise die sich teilweise etabliert hat. Darunter ist zum Beispiel zu verstehen, dass in einem gemanagten Gerät eine feste Funktionalität implementiert ist, die grundsätzlich in den Bereich SMFs eingeordnet werden kann, welche aber durch das Schreiben einer „1" an einer vom Hersteller vorgesehenen Position via SNMP aktiviert wird. Technisch ist dieses Vorgehen zwar möglich, jedoch außerhalb jeder Spezifikation und von multiplen kritischen Punkten behaftet (fehlender Rückkanal, Übertragbarkeit auf andere Hersteller, Verhaltensbeschreibung der pseudo SMF, etc.).

Durch den eher überraschenden Erfolg von SNMP und die damit einhergehende Verbreitung sind vor allem in der Mitte der 1990er Jahre - zu diesem Zeitpunkt gingen viele Experten tatsächlich noch davon aus, dass OSI-Management SNMP ablösen würde - einige Ansätze entstanden um SNMP und OSI-Management zu integrieren, z.B. [73] [21]. Keiner

der Ansätze hat dabei jedoch ein so breites Interesse erwecken können, dass er heute noch von praktischer Relevanz ist.

Bevor SNMP der gleichen Bewertung wie zuvor in Abschnitt 4.1.2 das OSI-Management unterzogen wird, nimmt der folgende Abschnitt eine technologische Beschreibung vor und versucht die Herausforderungen zu diskutieren, denen sich SNMP heute in der Praxis gegenüber sieht. SNMP stößt heute in vielerlei Hinsicht an seine Grenzen.

4.2.1 Technologische Einordnung

Es ist im vorangegangenen Abschnitt schon angeführt worden, dass SNMP ein sehr pragmatischer, auf Einfachheit ausgelegter Standard ist. Das bezieht sich auf alle Teile des Management-Modells. Es werden genau genommen überhaupt nur zwei der vier Modelle explizit adressiert: das Informations-Modell und das Kommunikations-Modell. Das Organisationsmodell ist (ursprünglich) sehr einfach, ein Funktionsmodell existiert de facto nicht. Dennoch soll auch SNMP, analog zum OSI-Management, anhand der vier Management-Modelle eingeführt werden.

Korrekterweise muss angeführt werden, dass der Begriff Standard hier nicht immer vollständig richtig ist, da nur wenige der RFCs wirklich einen finalen Stand haben, einige sind seit mehr als zehn Jahren im Entwurfsstatus, dennoch sind sie in Nutzung und weit verbreitet. RFCs werden allgemein als *Internetstandards* bezeichnet, unabhängig, ob sie diesen Status schon vollständig erreicht haben.

Das SNMP-*Organisationsmodell* wie in [8] ist auf den ersten Blick ähnlich dem OSI-Pendant. Es definiert Manager und Agenten, die miteinander kommunizieren. Allerdings ist es so, dass, betrachtet man beide Komponenten im Vgl. zum OSI-Management, der Anspruch an Einfachheit zum Vorschein kommt. Ein Agent stellt Informationen zur Verfügung bzw. nimmt diese entgegen und schreibt/setzt sie, der Manager präsentiert sich als Gegenstück. Mehr ist im SNMP-Organisationsmodell nicht vorgegeben. Erst ab Version 2 wurde mit der Manager-Manager-Kom-

munikation die Möglichkeit der Hierarchisierung eingeführt. Wie Manager und Agent implementiert werden, ist nicht definiert, lediglich, welche Dienste angeboten werden müssen.

Tabelle 4 SNMP Management-Operationen

Operation/ Dienst	Bedeutung
SNMPget	Abrufen von Management-Informationen, exakte Adressierung notwendig.
SNMPset	Modifikation von Management-Informationen.
SNMPgetnext	Abrufen von Management-Informationen, liefert nächste gültige Information.
SNMPgetbulk	Ruft eine gegebene Liste von Management-Informationen ab, erst ab SNMPv2c verfügbar.
SNMPresponse	Antwort auf die vorangegangenen Operationen.
SNMPtrap	Notifikation, die vom Agent ausgesendet werden kann. Die vollständige Integration ist erst ab SNMPv2c gegeben.
SNMPinform	Quittierte Notifikation, primär in der Manager-Manager-Kommunikation, ebenfalls mit SNMPv2c

An dieser Stelle kommt wieder das *Kommunikationsmodell* zum Tragen. Auch hier ist Einfachheit das Hauptbeschreibungsmerkmal. Anders als OSI setzt SNMP auf den verbindungslosen Transport, mit allen Vor- und Nachteilen. Alle von SNMP angebotenen Dienste sind in Tabelle 4 dargestellt.

Für die Integration zwischen SNMP-Versionen (vor allem Version 1 und Version 2) sind Proxies in [74] definiert. Weiterhin werden Proxies, die über die Integration zwischen SNMP-Versionen hinausgehen, in [75] adressiert. Allerdings geschieht dies auf einer sehr grundsätzlichen Ebene. In diese zweite Kategorie Proxies zählen jene, die die Integration nicht SNMP-fähiger Geräte (etwa Hubs) aber auch nicht SNMP-fähiger Protokollwelten für SNMP umsetzen. Als Beispiel im Kontext der

Automation seien [45], [46] mit der Integration von SNMP in Profibus angeführt.

Das *Funktionsmodell* ist das am schwächsten ausgeprägte der vier Management-Modelle im SNMP-Kontext. Es ist grundsätzlich keines definiert. Es wird vielmehr stillschweigend davon ausgegangen, das SNMP bzw. die IETF FCAPS unterstützen. Es kann im Grunde nur aus dem praktischen Einsatz von SNMP darauf geschlossen werden, welche Bestandteile von FCAPS umgesetzt werden. Weiterhin könnte man einzelne MIBs bzw. Erweiterungen (bspw. RMON [76], [77]) jedoch grundsätzlich als SMFs, bzw. als Teilaspekte solcher, im OSI-Sinne auffassen.

Tatsächlich ist es so, dass das *Informationsmodel* [31], [78] von allen vier Management-Modellen das ist, welchem die meiste Aufmerksamkeit gewidmet wurde. SNMP bedient sich eines sehr einfachen, flachen Informationsmodells, das als SMI (Structure of Management Information) bzw. in aktueller Version genau genommen SMIv2 [78] (zusammen mit SNMPv2 eingeführt) bezeichnet wird. Analog zum OSI-Management wird das Informationsmodell als Management Information Base (MIB) bezeichnet. Tatsächlich unterscheidet sich SNMP-MIB von OSI-MIB in nahezu allen Belangen.

SNMP erlaubt nur die Definition einfacher Bezeichner-Werte-Paare mit dazugehörigen Datentypen. Es ist in SMI nicht möglich, komplexe Relationen zwischen einzelnen Werten auszudrücken. Die einzige Möglichkeit zur Strukturierung und zum Herstellen von Kontexten bieten Tabellen, die sich ebenfalls aus Bezeichner-Wert-Paaren zusammensetzen. Lediglich die Position im Registrierungsbaum entscheidet darüber, welche Position (Tabellenzelle) der jeweilige Wert einnimmt. Der Registrierungsbaum, aus dem OSI-Management bekannt, spannt ausschließlich eine Namensstruktur auf.

Für SNMP stehen bis zum heutigen Tage zahlreiche Definitionen – MIBs – zur Verfügung, darunter umfassende Standard-MIBs etwa für Netzwerkprotokolle, Übertragungstechnologien, zum Remote-Monitoring

(RMON [76], [77]) aber z.B. auch für die Manager-Manager-Kommunikation. Jedem Anwender steht es frei Erweiterungen vorzunehmen und diese in eigene MIBs zusammenzufassen, dafür ist lediglich ein global einzigartiger Einstiegspunkt innerhalb des Registrierungsbaumes notwendig, der von der IETF vergeben wird. Was innerhalb dieses „Vendor Specific" Bereiches definiert wird, muss lediglich den grundsätzlichen SMI-Regeln gehorchen. Ebenfalls ist vollständig offen, wie eine MIB verteilt wird. Häufig muss sie gesondert vom Gerätehersteller bezogen werden. Diese Freiheit und die nicht vorhandenen objektorientierten Mechanismen führen zu einer massiven Redundanz. Es ist üblich, dass gleiche Informationen neu definiert werden und dadurch mehrfach gehalten werden, nur, um sie in einen Kontext setzen zu können. Heute ist ein Zustand erreicht, in dem das sehr einfache Informationsmodell nur noch schwierig zu handhaben ist, da Vielfalt und Verteilung, gerade in komplexen Systemen, hohe Anforderungen an den „menschlichen" Benutzer stellen.

Nachdem die wesentlichen Aspekte des SNMP nun kurz eingeführt wurden, wird im folgenden Abschnitt die Bewertung bzw. Einordnung dieser Technologie anhand der aufgestellten Kriterien vorgenommen.

4.2.2 Bewertung

Die Bewertung wird anhand der in Abschnitt 3.2 aufgestellten Tabelle durchgeführt und hat somit direkten Bezug zu Herausforderungen (Abschnitt 2.5), wie man sie heute im System-Management in der Automation wiederfindet. Die jeweils getroffene Bewertung wird kurz begründet, um den Bezug zu den im vorangegangenen Abschnitt ausgeführten technologischen Eigenheiten zu verdeutlichen.

Tabelle 5 Bewertung Simple Network Management Protocol

Kriterium	Begründung	Bewertung
Plattformbindung	SNMP ist heute auf allen Plattformen verfügbar.	2 ↑

Kriterium	Begründung	Bewertung
Durchdringung	SNMP ist de facto Standard in Automation und Unternehmen.	2
Mächtigkeit	SNMP bietet zwar eine Reihe von Standard-MIBs, diese decken aber nie Spezifika ab. Dies erfolgt in herstellerspezifischen MIBs, die nicht zum Basis-Informationsmodell gezählt werden. Durch das Fehlen von objektorientierten Prinzipien tragen solche Herstellererweiterungen nicht zur Mächtigkeit des Basis-Informationsmodells bei.	1
Erweiterbarkeit	Erweiterungen am Informationsmodell lassen sich in herstellerspezifischen Lösungen einfach realisieren. Allgemeingültige Erweiterungen, die Rückwirkung auf das Basis-Informationsmodell haben, sind kaum möglich.	1
Objekte	SNMP-Objekte sind einfache Bezeichner-Werte-Paare, die keine weiterführenden Eigenschaften besitzen. Hinzu kommt, dass SNMP-Objekten ohne vorhandene Beschreibung (textuell im Beschreibungsteil einer MIB) in der Regel keine Semantik zugeordnet werden kann.	0
Beziehungen	SNMP erlaubt es nicht, Beziehungen zwischen Objekten auszudrücken.	0 ↑
Technologiebindung	Obwohl SNMP in der Praxis hauptsächlich in IP-basierten Netzwerken – welche einen immer noch wachsenden Anteil bilden - eingesetzt wird, ist es nicht auf diese limitiert. Für Ethernet basierte Automatisierungsnetze stellt dies ohnehin keine Einschränkung dar.	2
Flexibilität	Durch das sehr einfache Informationsmodell, welches im Wesentlichen nur für das Netzwerk-Management geeignet ist,	1

Kriterium	Begründung	Bewertung
	sowie durch die nicht sehr ausdrucksstarken Objekte wird die Flexibilität eingeschränkt.	
Integration	Nach der Festlegung in 3.2 (Integration mittels Informationsmodell) ist dieses Kriterium in SNMP nur mit erheblichen Abstrichen erfüllbar. SNMP lässt es zum einen aber grundsätzlich offen, auf Basis welcher Informationsquellen ein Agent seine MIB instanziiert, zum anderen ist der Umweg über Proxies möglich.	1 ↑
Einheitlichkeit	SNMP ist auf sieben Dienste (siehe Tabelle 4) beschränkt, mit denen alle Operationen an MOs durchgeführt werden. Andere Zugriffe sind nicht möglich.	2
Durchgängigkeit	SNMP kann grundsätzlich zum Management beliebiger Hard- oder Software eingesetzt werden. Als (nicht ganz ernstes) Beispiel sei an dieser Stelle auf [79] verwiesen. In dem genannten Internetstandard wird das Management einer Kaffeemaschine per SNMP beschrieben. Auch hier ist in der Praxis das ausdrucksschwache Informationsmodell der beschränkende Faktor, da das Management hoch komplexer Systeme über das ausschließliche Lesen und Setzen von Einzelwerten schwer umsetzbar ist.	1
Standardisierung	SNMP ist in den mehrfach referenzierten RFCs beschrieben und wird allgemeinhin als Standard akzeptiert. Der vorangegangene Abschnitt beschreibt, welche Aspekte in welchen RFCs behandelt werden.	2
Objektauswahl	Sofern der exakte Bezeichner des Zielobjektes bekannt ist, bietet SNMPget die Möglichkeit des direkten und freien Zugriffs.	1

Kriterium	Begründung	Bewertung
	Ist die genaue Bezeichnung nicht bekannt, bleibt nur das Durchlaufen eines Teiles des Registrierungsbaumes, in dem das Zielobjekt vermutet wird. Eine kontextabhängige Auswahl (vgl. Scoping) oder Selektion (vgl. Filtering), sowie eine hierarchische Aggregation auf Basis von Vererbung sind nicht möglich.	
Modellbereitstellung	Die Modellbereitstellung erfolgt, bis auf die Standard-MIBs, vollständig dezentral. Resultat ist in der Praxis, dass MIBs aufwendig beschafft werden müssen, nicht immer gelingt dies. So kommt es vor, dass die Semantik einzelner Informationen vor dem Endanwender verborgen bleibt. Es ist kein Mechanismus vorgesehen, um die MIBs, die dem Informationshaushalt im Agenten zugrunde liegen, aus dem Agenten selbst abzurufen.	0 ↑
Dynamik	SMI sieht keine Möglichkeit vor, während der Laufzeit neue MOs zu erzeugen. In der Regel kann nur zur Implementierung des Agenten festgelegt werden welche MOs im Betrieb zur Verfügung stehen.	0
Notifikationen	SNMP bietet mit SNMPtraps eine Möglichkeit der Agenten-initiierten Nachrichtenübermittlung. Traps bieten jedoch keine Möglichkeit der Quittierung, wodurch die Anwendung in kritischen Bereichen, gerade in der Automation, fraglich bleibt.	1
Methoden	Methoden bzw. deren Nutzung sind mit Standardmitteln nicht möglich. Erweiterungen wie ScriptMIBs [72] sind kaum verbreitet. Die häufig genutzten Seiteneffekte sind, vor allem herstellerübergreifend, nicht berechenbar.	0 ↑

Kriterium	Begründung	Bewertung
Transport	SNMP nutzt in der Regel einfache UDP-Datagramme, wie auch in der Automation üblich.	2
FCAPS	Praktisch werden durch SNMP FCP bedient, wobei dies in keinem SNMP-betreffenden Standard festgeschrieben ist.	(ohne Einfluss)
Konzeptkomplexität	Im Vergleich zu anderen Management-Ansätzen sind die Konzepte hinter SNMP sehr einfach, was maßgeblich zur großen Verbreitung geführt hat. Die starke Fragmentierung des Informationsmodells wirkt jedoch einschränkend.	(ohne Einfluss)
Anwendungsbereiche	SNMP ist auf allen Ebenen zu finden, in denen „Internet" genutzt wird.	(ohne Einfluss)

Im Mittel ergibt sich für die gewerteten Kriterien ein Wert von 1 für alle 18 Kriterien, betrachtet man die 5 Kriterien mit überdurchschnittlicher Relevanz gesondert, ergibt sich eine mittlere Bewertung von 0,6. SNMP eignet sich, obwohl es der meist genutzte Standard ist, generell nur durchschnittlich für das Anwendungsgebiet Automation, in den als besonders relevant eingestuften Kriterien sogar nur unterdurchschnittlich.

Die Bewertungen beziehen sich ausschließlich auf SNMP als Protokoll bzw. Management-Ansatz, in der Regeln wird ein NMS als Frontend (Manager) für SNMP genutzt. Die eingesetzten NMS liefern viele der mit „0" bewerteten Eigenschaften nach. Insofern ist die Qualität der durch SNMP in der Praxis erzielten Ergebnisse in aller Regel höher. Der Ansatz selbst ist damit aber unmittelbar auf umfangreiche, nicht standardisierte Werkzeuge Dritter angewiesen.

Eine sehr ausführliche Pro- und Contra-Bewertung von SNMP wird in [23] vorgenommen. Dabei wird unter anderem auf die Leistungs-

fähigkeit des Protokolls, des Informationsmodell, des Sicherheitskonzeptes und vieler weiterer Aspekte eingegangen. Für den Diskursbereich Automation sind die in [23] diskutierten Probleme von SNMP grundsätzlich auch relevant. Obwohl diese Veröffentlichung gut zehn Jahre zurückliegt, gelten alle dort angeführten Punkte auch heute noch nahezu uneingeschränkt. Den Erfolg von SNMP konnten auch die dort ausführlich beschriebenen Unzulänglichkeiten nicht negativ beeinflussen.

4.3 Web Based Enterprise Management

Das Web Based Enterprise Management (WBEM) ist nicht ein Standard zur Verwaltung von Entitäten in einer gemanagten Umgebung, sondern es ist vielmehr der Sammelbezeichner für eine ganze Gruppe von technischen Spezifikationen. Im Rahmen dieser Arbeit kann nicht auf alle Bestandteile eingegangen werden, einige wenige werden im Rahmen des folgenden Abschnittes eingeführt, da dies für eine objektive Bewertung zwingend ist, andere werden lediglich erwähnt.

Die ersten Entwicklungsbemühungen wurden 1996 von einem Konsortium unter anderem bestehend aus Microsoft, Cisco System und Intel zusammen mit der DMTF (Distributed Management Task Force) unternommen. 1999 wurde WBEM, zusammen mit allen zu diesem Zeitpunkt schon entwickelten Subspezifikationen, an die DMTF übergeben, die seit diesem Zeitpunkt alle zugehörigen Dokumenten und Spezifikationen verwaltet. Im Rahmen der Arbeiten zu WBEM wurde neben anderen Spezifikationen auch das Common Information Model (CIM) definiert. CIM zählt heute zu den wichtigsten Spezifikationen der DMTF und wird neben WBEM auch noch in einer Reihe weiterer DMTF Standards genutzt. WBEM und die meisten in diesem Kontext relevanten Spezifikationen, sind auch zum Zeitpunkt der Fertigstellung dieser Arbeit, knapp 20 Jahre nach den ersten Veröffentlichungen, noch überaus lebendig.

Über die Zeit wurde eine wachsende Anzahl an technischen Spezifikationen dem Rahmenstandard WBEM hinzugefügt, darunter Service Discovery [80], [81], Nutzung von Web- und RESTful-Services [82], [83], [84], [85], [86], [87] zum Austausch von Management Informationen,

Möglichkeiten zur Definition von Policies [88]. Darüber hinaus existieren noch viele angrenzende Standards und Spezifikationen die in gewisser Weise einen Bezug zu WBEM besitzen (Storage Management Initiative - Specification - SMI-S, Virtualization Management - VMAN und weitere).

Genau genommen ist CIM selbst gar kein Teil von WBEM, auch wenn es ursprünglich in diesem Zusammenhang entwickelt wurde. Tatsächlich schreibt [89] die Nutzung von CIM auch in keiner Weise vor. Aufgrund der Fülle an Spezifikationen die im Kontext von WBEM existieren scheint es notwendig die Reichweite dieses Abschnittes im Vorfeld zu begrenzen. Der grundsätzliche Funktionsumfang von WBEM wird dadurch nicht eingeschränkt, da die meisten Erweiterungen ohnehin nicht zwingend sind bzw. nicht als Kernkomponente betrachten werden. Alle folgenden Ausführungen beziehen die Spezifikationen „Generic Operations" [89], „CIM Operations over http" [90], „Represenation of CIM in XML" [91], „Common Information Model (CIM) Infrastructure (Version 2.7.0)" [92] sowie direkt angrenzende Spezifikationen, insbesondere die jeweils gültigen CIM Schemata, mit ein. Im Zusammenhang der genannten Spezifikationen, inklusive einiger weiterer, ist häufig von WBEM/CIM die Rede, diese Nomenklatur wird auch in dieser Arbeit übernommen.

An einigen Stellen wird im Verlauf dieser Arbeit nicht auf die aktuelle Version bestimmter Spezifikationen zurückgegriffen, da sie zum Zeitpunkt der Arbeiten an einigen wesentlichen Abschnitten noch nicht zur Verfügung standen. Weiterhin ist es so, dass es teilweise einige Jahre in Anspruch nimmt, bis Änderungen und Erweiterungen, die von der DMTF ratifiziert wurden, auch ihren Weg in verfügbare Werkzeuge finden. Insbesondere sind davon das Integrieren weiterer Kommunikationsprotokolle, größere CIM-Schema-Erweiterungen und CIM Metaschema-Änderungen betroffen. Die im Kontext dieser Arbeit herangezogenen Versionen der Spezifikationen werden jeweils mit angegeben.

An dieser Stelle soll aber keineswegs der Eindruck entstehen, dass WBEM/CIM von regelmäßigen Änderungen betroffen sind, welche die Kompatibilität mit vorangegangenen Versionen nachhaltig beeinträch-

tigen. Ein solches Vorgehen wäre im Bereich von Management-Technologien nicht tragbar. Vielmehr ist es so, dass Komponenten, wie in der Softwaretechnik nicht unüblich, als „depricated" gekennzeichnet werden, in aller Regel aber mindestens bis zum nächsten „Major Release" unbeeinträchtigt in der Spezifikation enthalten bleiben. Es handelt sich dabei jeweils um einen Zeitraum von mehreren Jahren. Die neueste Version des CIM Metamodells [93] beispielsweise hat erst nach mehr als zehn Jahren (CIM Spezifikation 2.0 1999) grundlegenden Änderungen vorgenommen. Nach Aussage von Experten ist vor dem Jahr 2020 keine breite Unterstützung, etwa in etablierten Open Source Servern, der Version 3 des CIM Metaschemas zu erwarten. Ähnliche Aussagen treffen für andere Teile der Spezifikation und auch der eigentlichen CIM Schemata zu.

WBEM/CIM sind weiter verbreitet als gemeinhin angenommen. So ist beispielsweise fast jede Version des Desktopbetriebssystems Microsoft Windows mit WBEM/CIM Funktionalitäten ausgerüstet. Allerdings sind diese eher unter der Bezeichnung WMI (Windows Management Instrumentation) bekannt. Im Umfeld der Linux-Systeme sind verschiedene WBEM Dienste für nahezu alle Distributionen verfügbar, hier ist auch die noch recht neue Initiative OpenLMI [94] zu nennen. WBEM wird generell von verschieden Software und Hardwarelieferanten unterstützt, aktiv mitentwickelt (siehe dazu [95]) und ist in namenhaften Produkten bzw. Produktserien integriert. Generell stehen WBEM/-CIM, bzw. die notwendigen Infrastrukturen, für eine Reihe von Plattformen zur Verfügung: Linux, Unix, Mac, Windows, OpenVMS, zOS, VxWorks und weitere. In den Microsoft Windows (Desktop und Server) Betriebssysteme existiert, neben den freien Implementierungen, auch eine Umsetzung, die direkt von Microsoft stammt und unter der Bezeichnung „Windows Management Instrumentation" (WMI) bekannt ist. WMI ist dabei nicht „Protokollkompatibel" mit WBEM, vertraut jedoch auch auf CIM als Informationsmodell. VxWorks als Echtzeitsystem für eingebettete Geräte zeigt, dass WBEM durchaus auch abseits der reinen IT Installationen vorhanden ist.

Im Bereich der Automation generell hat WBEM/CIM bislang wenig Bedeutung, es existieren jedoch erste Arbeiten und Projekte [96], [97], [98] im Kontext der Verkehrsleitsysteme und auch im Bereich der industriellen Automation gibt es akademisch [99], [100], [101] erste Aktivitäten die veröffentlicht wurden, sowie erste Industrieprojekte [102], die von der vorliegenden Dissertation beeinflusst und vorangetrieben wurden.

Im Rahmen des nächsten Abschnittes liegt der Fokus also klar auf den „ursprünglichen" Kernkomponenten von WBEM (CIM abgebildet in XML über HTTP) inklusive CIM (v.2.7), die gegenwärtig auch die größte Verbreitung haben. Nachdem CIM schon mehrfach erwähnt wurde wird im folgenden Abschnitt, im Rahmen der Ausführungen zum Informationsmodell, detaillierter darauf eigegangen.

4.3.1 Technologische Einordnung

Um WBEM/CIM technologisch näher beschreiben zu können ist es notwendig, zunächst noch einmal klar zu stellen, dass WBEM und CIM vom Grunde her zwei vollkommen unabhängige Technologien sind. WBEM kann auch ohne CIM genutzt werden, CIM auf der anderen Seite ist erst einmal nicht mehr als ein Informationsmodell, das auch anderweitig angewendet werden kann (vgl. z.B. Abschnitt 4.6 zu WIMA). In der Praxis werden WBEM und CIM natürlich häufig als zusammengehörig betrachtet. Würde man den Versuch unternehmen die managementmodellrelevanten Verantwortlichkeiten zwischen WBEM und CIM aufzuteilen so wäre das Organisationsmodell und das Kommunikationsmodell Bestandteil von WBEM; das Informationsmodell und das Funktionsmodell würden durch CIM repräsentiert.

Welche Spezifikationsdokumente im Kontext der Betrachtungen liegen wurde bereits im einleitenden Teil zu WBEM/CIM kurz angeführt, dass damit bei weitem nicht alle Aspekte erfasst sind ist offensichtlich. Management-Profile etwa machen mittlerweile einen nicht unerheblichen Anteil an Spezifikationsdokumenten aus. Diese Profile sollen

Anwendern ein einfaches Mittel bieten, um sich in den sehr umfangreichen CIM-Schemata zu orientieren und einen vergleichsweise schnellen Einstieg ermöglichen, wenn es um konkrete Management-Aufgaben geht. Zum Zeitpunkt des Entstehens dieses Abschnitts waren über 75 Profile verfügbar, die ein Spektrum von Lüfter-, über Diagnose-, über Netzwerkdienst- bis hin zu Datenträger- und Server-Profilen abdecken.

Das *Organisationsmodell* ist in WBEM/CIM wahrscheinlich das am wenigsten ausgeprägte [23]. Lediglich in [89] werden einige Prinzip bedingte Anforderungen definiert. Generell kann man aber von einer Architektur und Organisation wie z.B. in [61], [103] beschrieben ausgehen. Wie in Abbildung 8 dargestellt besteht eine WBEM Architektur grundsätzlich aus drei Ebenen: dem Client (Manager im SNMP Kontext) auf oberster Ebene (i), häufig wird dies ein NMS bzw. eine anderweitige Anwendungssoftware sein; auf der zentralen Ebene (ii) befindet sich der so genannte CIM-Object-Manager (CIMOM), er fungiert als Broker zwischen Client und Provider bzw. Informationsmodell, welches wiederum im Repository gehalten wird.; die unterste Ebene (iii) bilden

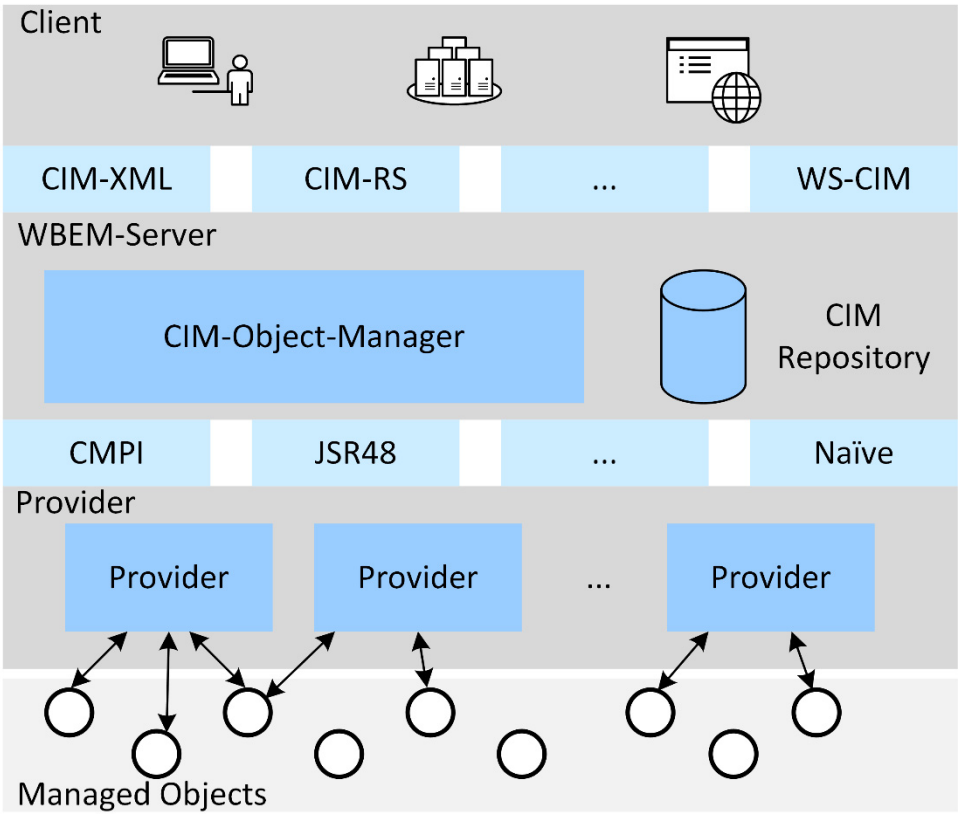

Abbildung 8 Grundstruktur WBEM

die Provider, sie sind die Verbindung zwischen den Managed Elements (vgl. Managed Objects in OSI und SNMP) des Informationsmodells und realen Entitäten. Dies ist eine vereinfachte Darstellung. Detaillierte Ausführungen zur Architektur sind etwa in [61] gegeben. Aus Sicht eines Clients handelt es sich auch bei WBEM um eine Client-Server Architektur, bei der der CIMOM als Server fungiert.

WBEM kann nahezu beliebig skaliert werden, allerdings ist dieser Sachverhalt nicht spezifisch in einer der Spezifikationen festgehalten. In [23] werden dieser Sachverhalt und seine Implikationen diskutiert. Die Schnittstellen zwischen CIMOM und Providern sind, ganz im Gegensatz zur wichtigeren Schnittstelle zwischen CIMOM und Client, ebenfalls nicht Bestandteil der DMTF Spezifikationen. Allerdings haben sich hier einige Provider Interfaces entwickelt die gut definiert, verbreitet und allgemein akzeptiert sind, allen voran CMPI (Common Manageability Programming Interface) [104]. Ähnlich wie z.B. OPC UA (vgl. 4.4.1) muss auch WBEM, bezogen auf den CIMOM, nicht vollständig von einer Implementierung umgesetzt werden. Zwingend ist nur, dass die Schnittstelle zwischen CIMOM und Client auf alle definierten Anfragen in einer korrekten Weise reagiert.

An diesem Punkt setzt das *Kommunikationsmodell* an. Im einleitenden Teil wurde bereits kurz auf die im WBEM/CIM Umfeld vorhandene Vielfalt an Methoden zur Übertragung der Management-Informationen zwischen CIMOM und Client eingegangen. Es existieren auf Seiten der DMTF Abbildungen für die entsprechenden Encodierungen [82], [86], [91], der Transport erfolgt also jeweils mittels http, jedoch in verschiedenen Diensten bzw. Kapselungen.

Interessanter im Rahmen der Ausführungen zum Kommunikationsmodell ist die Betrachtung der durch WBEM angebotenen Dienste. Je nach letztlich gewählter Art der Übertragung unterscheidet sich die technische Ausprägung, die Funktionalitäten bleiben jedoch erhalten. Die DMTF spezifiziert, diesem Sachverhalt folgend, generische Operationen [89]. Tabelle 6 stellt die knapp 30 Operationen dar, mit deren Hilfe mit einem

CIMOM interagiert werden kann, es sind keine weiteren Dienste notwendig.

Tabelle 6 Wichtige generische Operation in WBEM

Operation / Dienst	Bedeutung
Get-/Modify-/Create-/Delete-Instance	Mit diesen Diensten wird das Lesen, Ändern, Erzeugen oder Löschen von Instanzen einer Klasse des Informationsmodells durchgeführt. Voraussetzung ist, dass eine spezifische Instanz bereits mit ihrem Namen (siehe unten) identifiziert wurde.
EnumerateInsatnce(s)/(Names)	Liefert alle Instanzen einer Klasse, bzw. Ausschließlich deren Namen
Associator(s)/(Names)	Durch diesen Dienst werden zu einer gegebenen Instanz alle Instanzen geliefert, die mit dieser Assoziiert sind, bzw. ausschließlich deren Namen.
Reference(s)/(Names)	Dieser Dienst liefert alle assoziativen Instanzen die von einer gegebenen Instanz referenziert werden.
Invoke(Static)Method	InvokeMethod führt, auf Basis des Namens der Instanz, die in der zugehörigen Klasse definierte Methode aus
Get/-Modify-/Create-/Delete-Class	Diese Dienste erlauben das Manipulieren von Klassen.
EnumerateClass(es)/(Names)	Liefert alle Klassen eines Namensraumes incl. Pfad, bzw. nur die Namen der Klassen.
AssociatorClass(es)/(Paths)	Liefert alle Klassen die mit einer gegebenen Klasse assoziiert sind, bzw. nur ihre Pfade.
ReferenceClass(es)/(Paths)	Liefert die assoziativen Klassen die eine gegebenen Klasse referenzieren, bzw. nur deren Pfade.
Get-/Delete-/Modify-/Create-/Enumerate-QualifierType(s)	Diese Dienste erlauben das Auflisten bzw. Manipulieren der Properties (siehe Informationsmodell unten) einer Klasse.

Die aufgeführten Operationen werden auch als intrinsische Methoden bezeichnet, während das Aufrufen von Methoden, via InvokeMethod, die durch das Informationsmodell definiert werden von extrinsischen Methoden gesprochen wird.

Ergänzend sei an dieser Stelle noch erwähnt, dass in der Version 1.4 der Spezifikation [90] ein Teil dieser Operationen auf „Depricated" gesetzt wurde, zum Vorteil der Open und Pull Verfahrensweise, die auch schon in [89] Version 1.0.2 beschrieben werden. Aufgrund der schon erläuterten „Depricated"-Verfahrensweise der DMTF und letztlich auch begründet im prinzipiell identischen Funktionsumfang wird hier darauf verzichtet eine Detailbetrachtung durch zu führen.

In der Auflistung der WBEM Operationen in Tabelle 6 fehlt jeder Bezug zu Indication, dem WBEM Event-Mechanismus. Tatsächlich sind Indications nicht Teil der Spezifikation [89], da keine explizite Operation für Indications existiert. Die Konfiguration einer Indication, mit allen zu diesem Mechanismus gehörenden Schritten, wird von den in Tabelle 6 aufgelisteten Operationen implizit mit abgedeckt. Dies ist vor allem darin begründet, dass WBEM-Server ihre eigene Konfiguration, also auch z.B. die Adressaten von Indications, ebenfalls in einer CIM-Struktur verwalten. Der Transport der eigentlichen Indication vom CIMOM zum so genannten Listener erfolgt, im Kontext von „CIM Operation over http" [90], über spezielle XML Nachtichten [90], [91]. Für eine detaillierte Beschreibung sei auch hier wieder auf [61] verwiesen. Alternativ existiert ein sehr umfangreiches Profil [105] welches alle Sachverhalte zum Thema Indications in WBEM beinhaltet oder referenziert.

Was das *Funktionsmodell* in WBEM/CIM betrifft so muss hier konsequenter Weise auch die Aussage gelten das keines definiert ist, denn eine Spezifikation die entsprechend Position bezieht existiert nicht. In der Tat ist es aber so, dass Bezüge zu FCAPS von Seiten der DMTF offiziell hergestellt werden [106]. Zusammenarbeitsbekundungen zwischen DMTF und ITU-T für die Thematik „FCAPS interfaces" existieren im Rahmen der „Next Generation Network Management Focus Group"

[107]. Praktisch ist WBEM/CIM in der Lage FCAPS vollständig abzudecken und geht sogar noch darüber hinaus, denn es werden explizit noch die Abstraktion und Dekomposition von Diensten und Geschäftsprozessen [106] mit einbezogen.

Es lassen sich noch weitere Parallelen zwischen den mit FCAPS in Zusammenhang stehenden SMFs (vgl. 4.1.1) herstellen. Einige DMTF Profile (DSPs 1000 bis 1117 [108]) haben in ihrer Aufgabendefinition sehr starke Bezüge zu SMFs. Als ein Beispiel sei hier das „Common Diagnostrics Profile" [109] genannt. Obwohl FCAPS in keiner zu WBEM/CIM gehörenden Spezifikation explizit erwähnt wird, ist es doch ein fester Bestandteil der Standards. Auch das *Informationsmodell* – CIM – bietet alle Voraussetzungen bzgl. Flexibilität und Funktionsumfang.

CIM, das *Informationsmodell*, ist ein vollständig objektorientiertes Modell. Es besteht auf oberster Ebene aus zwei Komponenten dem Meta Schema [92] und dem eigentlichen CIM-Schema. Organisatorisch sind für das Schema drei Gruppen definiert: das Core Schema, die Common Schemata und die Extension Schemata. Die Core und Common Schemata werden direkt von der DMTF verwaltet und beinhalten neben einer sehr groben Struktur – Core Schema – Schemata für eine Vielzahl von Teilbereichen des System-Managements (Netzwerk, Geräte, Datenbanken, Metriken und viele mehr). Als Extension Schemata werden alle die Teile des Modells bezeichnet, die durch Nutzer ergänzt oder erweitert wurden, dies kann z.B. notwendig sein, um Firmenspezifika in CIM zu integrieren. Es ist nicht ausgeschlossen das Extension Schemata in Common Schemata teilweise oder ganz übergehen. Dies ist durch Erfüllung bestimmter Voraussetzungen und letztlich durch die Ratifizierung durch die DMTF möglich.

CIM wird neben WBEM noch in einer Reihe anderer Standards genutzt. Das ist vor dem Hintergrund, dass es wohl das umfassendste und ausdruckstärkste Informationsmodell ist, das einem breiten Nutzerkreis zur Verfügung steht und von diesem auch genutzt wird, gut nachvollziehbar. Zwei wesentliche Stärken der anderen beiden im Netzwerk- und System-Management bekannten Informationsmodelle wurden in CIM zusammengefasst [106]: zum einen das mächtige explizit

beschriebene Metamodell aus dem OSI-Management (vgl. 4.1.1), zum anderen die große Anzahl an Standard- bzw. Basis-Komponenten des Informationsmodells wie sie durch die in SNMP bekannten Standard-MIBs (vgl. 4.2.1). CIM bietet an dieser Stelle beides. Diese Mächtigkeit hat natürlich einen Preis – Komplexität. Zum Zeitpunkt des Entstehens dieses Abschnittes bestanden die CIM Core und Common Schemata insgesamt aus über 1700 Klassen. Profile (DSPs ab 1000) helfen jedoch diese Komplexität für einen Anwender, sofern seine spezifische Management-Aufgabe durch ein oder mehrere Profile abgedeckt wird, zu minimieren.

CIM unterstützt schon in Version 2 des Metaschemas alle wesentlichen objektorientierten Eigenschaften. In Version 3 wurde das Metaschema noch einmal um mächtige Eigenschaften wie etwa die Object Constrain Language (OCL) erweitert. Aus den im einleitenden Teil bereits ausgeführten Gründen wird Version 3 hier nicht in Betracht gezogen. Zum Zeitpunkt der Arbeiten an wesentlichen Teilen der Arbeit war Version 3 des Metaschemas noch nicht verfügbar, eine Unterstützung in verfügbaren Werkzeugen ist bis zum Ende der Arbeiten an dieser Dissertation nicht absehbar gewesen.

Wie auch schon die beiden wahrscheinlich bekanntesten, wenn auch vollkommen verschiedenen, Netzwerk- und System-Management-Ansätze – OSI-Management und SNMP – soll nun auch WBEM/CIM anhand der aufgestellten Kriterien genauer eingeordnet werden.

4.3.2 Bewertung

Die Bewertung wird anhand der in Abschnitt 3.2 aufgestellten Tabelle durchgeführt und hat somit direkten Bezug zu Herausforderungen (Abschnitt 2.5), wie man sie heute im System-Management in der Automation wiederfindet. Die jeweils getroffene Bewertung wird kurz begründet, um den Bezug zu den im vorangegangenen Abschnitt ausgeführten technologischen Eigenheiten zu verdeutlichen.

Tabelle 7 Bewertung WBEM und CIM

Kriterium	Begründung	Bewertung
Plattformbindung	WBEM/CIM bzw. eng verwandte Umsetzungen sind heute für sehr viele Plattformen verfügbar. Dies reicht von Desktop- und Server-Systemen bis hin zu eingebetteten Systemen in verschiedensten Anwendungsbereich.	2 ↑
Durchdringung	WBEM/CIM ist im Unternehmensbereich zweifelllos angekommen [110], aber auch in der Automation gibt es wie in 4.3.1 angeführt erste Bemühungen, nicht zuletzt sind Microsoft Windows Produkte Satus Quo somit auch WMI. In beiden Bereichen besteht jedoch sicherlich noch Wachstumspotential. WBEM/CIM ist im direkten Vergleich mit SNMP dennoch weniger verbreitet.	1
Mächtigkeit	CIM ist gegenwärtig das mächtigste bekannte Informationsmodell.	2
Erweiterbarkeit	Erweiterungen an CIM sind fest vorgesehen und durch den vollständigen objektorientieren Ansatz jeder Zeit möglich. Ebenfalls besteht die Möglichkeit, dass Erweiterungen in den Status von „Standards" erhoben werden, in dem sie Teil eines Common Schemas werden.	2
Objekte	Objekte in WBEM/CIM können eine nahezu unbegrenzte Mächtigkeit besitzen und sind klar strukturiert.	2
Beziehungen	Durch die vollständige Objektorientierung erlaubt CIM schon grundsätzlich das Abbilden von Beziehungen zwischen Objekten. Dazu existiert noch eine Reihe von so genannten Qualifier [92], um Beziehungen weiterreichend zu annotieren.	2 ↑

Kriterium	Begründung	Bewertung
Technologiebindung	WBEM/CIM hat im Rahmen der Definitionen in Abschnitt 3.2 keine Bindungen an Technologien, die für die Automation unbekannt sind und somit als kritisch angenommen werden müssen.	2
Flexibilität	Hier fließen die große Mächtigkeit des Informationsmodells und die Ausdrucksstärke der Objekte zusammen. WBEM/CIM kann für jeden Management-Belang eingesetzt werden oder unkritisch um die jeweiligen Belange erweitert werden.	2
Integration	In CIM existiert das Konzept der MappingStrings, die direkt, also auf Ebene des Informationsmodells, die Integration von Informationen aus anderen Modellen ermöglichen. Wie mit diesen MappingString am Ende verfahren wird, bleibt jedoch dem Anwender bzw. Entwickler der Management-Lösung überlassen. Ein Proxy-Konzept lässt sich in WBEM/CIM ohne weiteres umsetzte, für den CIMOM ist es gleichgültig, wie die eigentlichen Informationen durch einen Provider beschafft werden. Jeder Provider kann also ein Proxy sein.	2 ↑
Einheitlichkeit	Tabelle 6 fasst die relevanten Zugriffsmöglichkeiten zusammen. Es existieren, abgesehen von Verfeinerungen, keine weiteren. Egal was im Fokus der jeweiligen Management-Aufgabe steht, es kommen aus Sicht des Clients immer die gleichen Mechanismen zum Einsatz.	2
Durchgängigkeit	Die DMTF schränkt den Einsatz von WBEM und CIM nicht ein. Auch in der Praxis homogenisiert WBEM/CIM das Management, ein Lüfter wird grund-	2

Kriterium	Begründung	Bewertung
	sätzlich mit den gleichen Mitteln gemanagt wie ein komplexes Datenbanksystem.	
Standardisierung	Wie im Einleitenden Abschnitt zu WBEM und CIM beschrieben ist die DMTF hier das standardisierende Gremium, im Wesentlichen erfolgt die Standardisierung also analog wie bei SNMP (dort durch die IETF) oder OSI/TNM (durch die ITU-T).	2
Objektauswahl	Objekte können vollkommen frei gewählt werden. Dafür stehen gleich mehrere Möglichkeiten zur Verfügung. Neben den Methoden aus Tabelle 6 (z.B. Enumerate und EnumerateNames) existieren auch noch Anfragesprachen, die eine Selektion ähnlich wie bei Datenbanken bieten.	2
Modellbereitstellung	Die Basiskomponenten des Informationsmodells werden zentral durch die DMTF zur Verfügung gestellt und gepflegt. Über Methoden (vgl. Tabelle 6) zum enummerieren von Klassen und Properties können auch bislang unbekannte Modellelemente aus einem Laufenden CIMOM abgefragt werden.	2 ↑
Dynamik	WBEM/CIM bietet auch hier viele Freiheiten. Klassen können auch dynamisch angelegt und manipuliert und gelöscht werden, gleiches gilt für Objekte.	2
Notifikationen	WBEM/CIM bietet grundsätzlich zwei Gruppen von Notifikationen (Indications). Notifikationen können sowohl von externen Ereignissen abgeleitet werden (Ereignisse im/am realen gemanagten Objekt), wie auch auf Basis von Änderungen am Informations-	2

Kriterium	Begründung	Bewertung
	haushalt (Instanzen im Informationsmodell) aus dem CIMOM selbst emittiert werden.	
Methoden	WBEM/CIM bietet unbegrenzt mächtige Methoden.	2 ↑
Transport	Abgesehen davon, dass WBEM mehrere Möglichkeiten zum Datentransport bietet, ist auch der „Standard-Weg" über [90] http in der Automation nicht mehrfremd.	2
FCAPS	Nimmt man [106] als Grundlage kann WBEM/CIM FCAPS vollständig abdecken und geht in einigen Belangen sogar noch darüber hinaus.	(ohne Einfluss)
Konzeptkomplexität	WBEM/CIM stellt sich zweifellos als sehr komplexer Ansatz dar. Dies ist nicht mit einer Vielzahl an Standards o.Ä. zu begründen sondern ist auf das massiv komplexe Informationsmodell zurück zu führen.	(ohne Einfluss)
Anwendungsbereiche	WBEM/CIM kann in sehr unterschiedlichen Domänen eingesetzt werden, die einfache Erweiterbarkeit setzt hier keine Grenzen. Gegenwärtig wird WBEM/CIM vorwiegend im Unternehmensbereich eingesetzt, wenn das ganzheitliche Management von komplexen Soft- und Hardwaresystemen im Fokus liegt.	(ohne Einfluss)

Im Mittel ergibt sich für die gewerteten Kriterien ein Wert von 1,9 für alle 18 Kriterien. Betrachtet man die 5 Kriterien mit überdurchschnittlicher Relevanz gesondert, ergibt sich eine mittlere Bewertung von 2. WBEM/CIM eignet sich uneingeschränkt als Management-Technologie für die Automation, sofern man ausschließlich die in dieser Arbeit aufgestellten Kriterien in Betracht zieht.

Auch wenn die Bewertung von WBEM/CIM auf den ersten Blick den Eindruck hinterlässt, als wäre eben diese Technologie vollkommen frei von Hindernissen, so müssen doch noch einige Punkte kritisch angeführt werden. Der schwerwiegendste Punkt ist wohl die Komplexität des Informationsmodells CIM. Man kann davon ausgehen, dass auch automationsspezifische Belange, auf einer abstrakten Ebene, schon in CIM enthalten sind. Die große Herausforderung ist es aber, diese abstrakten Belange in einem sehr umfangreichen Modell zu identifizieren, nur wenn dies gelingt, kann um Automationsspezifika erweitert werden – ein praktisches Problem. Die Option nur das CIM-Metamodell, ohne Core- und Common-Schemata, zu nutzen und ein vollständig neues Informationsmodell aufzubauen besteht und wird in Abschnitt 5.3 diskutiert.

Wie im einführenden Teil zu WEBM/CIM dargestellt wird zur Kommunikation vorwiegend XML über http genutzt. Ob diese Technologiekombination, die auf den ersten Blick den Eindruck hinterlässt, als würde sie im Kontrast zu den generellen „Lightweight"-Anforderungen der Automation stehen, den Anforderungen in Automatisierungsnetzen dennoch genügt wird in Abschnitt 5.1 diskutiert.

4.4 OPC UA

OPC (OLE (Object Linking and Embedding) for Process Control) wurde als Schnittstelle für den einheitlichen und feldbusunabhängigen Zugriff auf Informationen in automatisierungstechnischen Anlagen eingeführt. Gepflegt und entwickelt wird OPC von der OPC Foundation. Dies gilt auch für neue Versionen und Erweiterungen. OPC hat sich als de facto Standard für den Zugriff auf Informationen innerhalb von Automatisierungsnetzen etabliert und erfährt eine breite Unterstützung. In seiner ursprünglichen Form (Version 1.0 1996) basierte OPC auf Microsoft-COM/DCOM. Diese klassische Version von OPC spielt in dieser Arbeit keine Rolle.

OPC UA ist die Weiterentwicklung [111], [112], [113] des klassischen OPC, verfolgt im Grundsatz aber noch immer das gleiche Ziel, den

einheitlichen Zugriff auf Daten innerhalb eines heterogenen Automatisierungsnetzes, erweitert dieses aber entsprechend der gewachsenen Anforderungen bzgl. Integration in der Automation. OPC UA zielt auf die vertikale Integration über alle Ebenen der Automatisierungspyramide (vgl. Abbildung 2) ab. Die Methoden und Basistechnologien, die dafür zum Einsatz kommen, wurden deutlich modernisiert, Details dazu sind im folgenden Abschnitt zusammengefasst. OPC UA bietet heute moderne Technologien wie Webservices, objektorientierte Modelle und bezieht klar Position zum Thema Datensicherheit – all diese Themen sind nach wie vor keine Selbstverständlichkeit in der Automation. Man kann nicht nur sagen, dass OPC bzw. dessen Weiterentwicklung mit den in Abschnitt 1.3.2 beschriebenen technologischen Trends in der Automation einhergeht, sondern an einigen Stellen den generellen Entwicklungen in der Automation technisch und organisatorisch voraus ist. So ist OPC UA z.B. auf Plattformunabhängigkeit ausgelegt. Stacks und Spezifikationen sind für einen breiten, nicht beschränkten Nutzerkreis offen und nahezu kostenfrei, dass Einbringen von Erweiterungen der Spezifikationen durch externe Organisationen ist möglich und wird auch gelebt. Beispiele sind [114] und weitere sogenannte Companion Standards. Daneben gibt es noch andere Aktivitäten, die mit OPC UA in Verbindung stehen [115], [116], [117].

Wofür OPC UA heute genau steht ist in den Hintergrund geraten. Teilweise wird es nicht mehr als Abkürzung sondern vielmehr als Technologiebezeichnung verstanden. In der Literatur finden sich aber auch „OPen Connectivity Unified Architecture" und „Openness Productivity Collaboration Unified Architecture" als ausgeschriebene Varianten. OPC UA umfasst heute eine Reihe von Spezifikationen die als IEC Normen der Serien IEC 62541 zugänglich sind.

Für das Netzwerk- und System-Management wird OPC gegenwärtig nicht (direkt) eingesetzt, auch wenn dem technisch nichts im Wege steht, wie in folgenden beiden Abschnitten erörtert wird. Vollständig neu ist die Betrachtung von OPC (UA) im Kontext von Netzwerk- und System-Management aber dennoch nicht.

Eine der Kernkompetenzen von OPC war schon immer die Anbindung von HMIs (Human-Machine-Interface). Mittlerweile existieren viele Produkte, die OPC und SNMP-basiertes Netzwerk-Management vereinen und so die gemeinsame Darstellung von Prozessdaten (via OPC) und Netzwerkinformationen (via SNMP) in einem HMI gewährleisten. Dazu gibt es am Markt von verschiedenen Herstellern Lösungen ([118], [119], [120], [121]), um SNMP-fähige Feldgeräte vollständig in OPC zu integrieren. Auch der umgekehrte Weg, also das Einbinden von OPC entstammenden Daten in klassische Management-Werkzeuge (über SNMP als Middleware) wird von einigen Firmen beschritten [122]. OPC im Kontext von Netzwerk-Management zu betrachten ist also gebräuchlicher, als auf den ersten Blick vermutet werden mag. Auch wenn all diese Produkte auf eine Integration zwischen dem klassischem OPC und SNMP abzielen, so ist ein analoges Vorgehen für OPC UA vorstellbar. Die Möglichkeiten für die Integration zwischen OPC UA und anderen Informationsmodellen ist grundsätzlich gegeben und wird unter anderem in [123], [124] für die Abbildung zwischen OPC UA und IEC 61970 beschrieben.

Bevor auch OPC UA der, aus den vorangegangenen Abschnitten bekannten, Bewertung unterzogen wird, nimmt auch hier der folgende Abschnitt eine technologische Beschreibung vor. Für eine umfassende Einführung in die Thematik OPC UA sei noch einmal auf [111], [112], [113] verwiesen.

4.4.1 Technologische Einordnung

Wie schon die Modelle und Technologien in den vorangegangenen Abschnitten wird OPC UA wird anhand der vier Management Modelle beschrieben. Bei OPC UA handelt es sich offensichtlich nicht unmittelbar um einen Standard zum Management. Von Seiten der Organisation und Erweiterbarkeit ließe sich eine Nutzung für das System-Management durchaus vorstellen. Nicht zuletzt sind für OPC UA zwei der schon bekannten Modelle explizit definiert und standardisiert (Kommunikations- und Informationsmodell). Für die anderen beiden (Organisations-

und Funktionsmodell) lassen sich aus der Praxis einige Aussagen ableiten.

Das *Organisationsmodell* ist eines der beiden Modelle, die in OPC UA nicht explizit vorhanden sind. Die meisten grundsätzlichen organisatorischen Festlegungen sind in [125] getroffen. Demnach folgt OPC UA einem klassischen Client-Server-Modell, wobei auch Mischformen möglich sind, etwa für Gateways oder Proxies bzw. zur Hierarchisierung der Strukturen. Im Kontext von OPC UA entspricht der Client dem aus dem Netzwerk-Management bekannten Manager, der Server dementsprechend dem Agent. Auch bei der Abbildung der Realität im System finden sich Parallelen zum System-Management. Reale physische oder logische Entitäten werden durch Objekte repräsentiert die ihrerseits in einer, bei OPC UA als Adressraum bezeichnet, internen Informationsstruktur gehalten werden.

Ähnlich wie beim WBEM (vgl. 4.3.1) müssen Dienste (vgl. OPC UA Kommunikationsmodell unten) in OPC UA nicht vollständig implementiert werden [113]. Dadurch soll eine größtmögliche Flexibilität bzgl. der Zielplattformen erreicht werden, sodass OPC UA sowohl auf leistungsstarker Büro- und Unternehmenshardware, wie auch auf kleindimensionierten eingebetteten Systemen, implementiert werden kann (vgl. [126] OPC UA Server Profile).

Die OPC UA spezifizierenden Standards lassen bislang weitestgehend offen, wie OPC UA für große Systeme skaliert. Eine Server-Server-Kommunikation ist nur über die Client-Server-Mischform möglich, nutzt in diesem Fall dann aber natürlich die regulären OPC UA-Dienste. Allerdings erlaubt OPC UA explizit der Verteilung auf Datenebene. Dafür bietet das Informationsmodell bzw. die durch selbiges beschriebenen Objekte, die Möglichkeit Referenzen zu nicht lokalen Adressraumelementen zu halten. Auf diese Weise können, für den Nutzer transparent, Informationen welche durch ein Objekt repräsentiert werden über mehrere Adressräume aggregiert werden.

Das *Kommunikationsmodell* [127] ist in OPC UA explizit ausgeprägt. Wie schon erwähnt, ersetzt es das klassische OPC Modell vollständig. Dabei

werden zwei Kommunikationsinfrastrukturen definiert. Binär encodierte Übertragung für größtmögliche Performance und XML/Webservice für größtmögliche Interoperabilität.

OPC UA stellt eine feste, überschaubare Menge an einheitlichen Diensten für den Zugriff auf alle Daten zur Verfügung. Die Anzahl an Diensten ist jedoch umfangreicher als etwa beim OSI-Management oder bei WBEM, deshalb werden an dieser Stelle (Tabelle 8) nicht wie gewohnt die Dienste selbst sondern die gruppierenden Service-Sets, so wie in [125] definiert, kurz zusammengefasst und erläutert. Vollständig werden alle 38 gegenwärtig definierten Dienste in [128] beschrieben.

Auch bei den Service Sets verlangt OPC UA nicht nach Vollständigkeit, um dem Anspruch an die Implementierbarkeit auf vergleichsweiser schwacher Hardware nachzukommen. Ob ein spezifischer Server eine bestimmte Dienstgruppe unterstützt oder nicht ist durch sein Profil festgelegt [126]. OPC UA bietet komplexe Eventmechanismen (siehe MonitoredItem- und Sbscription-Service-Sets Tabelle 8) auf Basis des Publish-Subscribe-Konzeptes, die als Notifikationen im Sinne eines System-Managements aufgefasst werden können. In der Praxis werden Alarme und Zustände [130] in der Tat über diesen Event-Mechanismus übermittelt.

Tabelle 8 OPC UA Service Sets

Service-Set	Bedeutung
Discovery	Zum Auffinden von OPC UA Servern im System und Zugang zu Sicherheitskonfigurationen für Clients benötigt.
SecureChannel	Gruppe von Diensten, die zum Etablieren eines sicheren Kommunikationskanals [129] benötigt werden, typischer Weise werden die SecureChannel Dienste nicht vom Anwender sondern vom Stack genutzt.
Session	Diese Gruppe beinhaltet Dienste, die zum Einrichten einer Verbindung auf Anwendungsebene benötigt werden.
NodeManagement	Diese Gruppe von Diensten erlaubt das Bearbeiten des OPC UA Adressraumes selbst (ändern, hinzufügen/löschen von Knoten)

View	Diese Gruppe stellt Dienste zum Durchschreiten des Adressraumes bzw. von Untermengen des Adressraumes zur Verfügung.
Attribut	Von dieser Dienstgruppe werden Mittel zur Manipulation von Attribut, also Knoteninhalten, zur Verfügung gestellt.
Method	Stellt einen Dienst zur Parameterkommunikation und zum Aufrufen der für einen Knoten definierten Methoden zur Verfügung.
MonitoredItem	Diese Gruppe enthält Dienste zur Konfiguration von Eventmechanismen durch den Client. Events können auf Basis von Zustands- bzw. Wertänderungen im Adressraum des OPC UA Servers ausgelöst werden.
Subscription	In dieser Dienstgruppe sind Dienste definiert, die zur Konfiguration der Nachrichtenübertragung an Clients benötigt werden.
Query	Diese Gruppe von Diensten ermöglicht den direkten Zugriff auf Inhalte des Adressraumes, der Zugriff kann dabei auf Basis von übergebenen Kriterien detailliert werden.

OPC UA definiert grundsätzlich kein *Funktionsmodell* im Sinne der aus dem OSI-Management bekannten SMFAs. Aus den Einsatzszenarien lässt sich jedoch ableiten, welche der fünf SMFAs OPC UA implizit abdeckt. Wie auch schon bei SNMP werden Accounting und Security in keiner Weise adressiert. Es ist an dieser Stelle jedoch darauf hinzuweisen, dass Security sich hier ausschließlich auf das Management von Sicherheitsaspekten und nicht auf Sicherheitsmechanismen in OPC UA (z.B. Verschlüsselung der Kommunikation) selbst bezieht, diese sind durchaus vorhanden. Configuration Management ist möglich, steht aber zumindest gegenwärtig nicht im Fokus der Anwender, dies könnte sich aber durch aktuelle Bemühungen wie OPC UA Devices [114] und FDI [115], [116] ändern.

Ein Ziel bei der Entwicklung von OPC UA war es, nicht nur den herstellerunabhängigen, interoperablen Zugriff auf Daten zu realisieren, sondern die Informationsbeschreibung selbst interoperabel und austauschbar zu gestalten. Das spiegelt sich natürlich im *Informationsmodell* wieder,

aber auch in Form des „View-Service Sets“ (Tabelle 8), welches es erlaubt Wissen über das Informationsmodell aus einem laufenden Server heraus zu extrahieren.

Grundsätzlich handelt es sich beim OPC UA Informationsmodell um ein objektorientiertes Modell. Einige Modellkonzepte werden bei OPC UA jedoch anders, als etwa in CIM, OSI oder allgemein in UML, interpretiert. So spricht der Standard von NodeClasses, die das Metamodell aufspannen. Nodes, sozusagen die erste Stufe der Instanziierung von NodeClasses, werden genutzt, um die, etwa aus UML, bekannten Klassen zu beschreiben. Diese Nodes haben jedoch eher den Charakter von Objektprototypen, da sie sowohl in beschreibenden (klassenähnlich) wie auch ich darstellender Form (objekteähnlich) vorkommen können. OPC UA besitzt ein eigenes Metamodell welches in [131] definiert wird, direkte Beziehungen zum allgemein bekannten UML bestehen nicht.

Die üblichen objektorientierten Konzepte, inklusive guter Erweiterbarkeit, sind in OPC jedoch uneingeschränkt vorhanden. Neben dem Metamodell [131] existiert ein Basisinformationsmodell [132], welches einige sehr allgemeingültige Definitionen enthält, etwa Basistypen und Einstiegspunkte in den jeweiligen Informationsraum. Dazu kommt mittlerweile eine Anzahl an offiziellen und teilweise auch bereits standardisierten Erweiterungen für spezifische Aufgaben. Daneben sind Hersteller oder Anwendungsfall spezifische Modelle möglich und erwünscht. OPC UA Informationsmodelle können eine beliebige Komplexität erreichen, die Verwendung ist aber keinesfalls zwingend. In der Theorie könnten Hersteller auch auf die Nutzung der vordefinierten Modellteile verzichten und nur auf das Metamodell zurückgreifen. Der angestrebten Interoperabilität wäre dies aber wahrscheinlich nicht zuträglich.

Gegenwärtig definiert OPC UA keine Integrationen von anderen Technologien auf Basis des Informationsmodells. Bezüglich der Integration mit bestehenden Technologien muss generell noch angeführt werden, dass OPC UA und der klassische OPC durch den Paradigen- und Technologiewechsel nicht direkt kompatibel sind. Allerdings werden von der OPC Foundation sowie von anderen Firmen Proxies und Wrapper für OPC/OPC

UA angeboten, um einen schnellstmöglichen Übergang zu gewährleisten.

Nachdem die wesentlichen Aspekte des OPC UA nun kurz eingeführt wurden, wird im folgenden Abschnitt die Bewertung bzw. Einordnung dieser Technologie anhand der aufgestellten Kriterien vorgenommen.

4.4.2 Bewertung

Die Bewertung wird anhand der in Abschnitt 3.2 aufgestellten Tabelle durchgeführt und hat somit direkten Bezug zu Herausforderungen (Abschnitt 2.5), wie man sie heute im System-Management in der Automation wiederfindet. Die jeweils getroffene Bewertung wird kurz begründet, um den Bezug zu den im vorangegangenen Abschnitt ausgeführten technologischen Eigenheiten zu verdeutlichen. Es wurde im vorangegangenen Abschnitt schon herausgestellt, das OPC UA nicht den Fokus auf das Management von Systemen legt, sondern primär die Handhabung von Prozessdaten als Ziel hat. Einige der bewerteten Kriterien erscheinen dadurch zu negativ bewertet, allerding muss noch einmal betont werden, dass hier spezielle Herausforderungen bzgl. des System-Managements adressiert werden.

Tabelle 9 Bewertung OPC UA

Kriterium	Begründung	Bewertung
Plattformbindung	OPC UA legt großen Wert auf Plattformunabhängigkeit, es sind heute Umsetzungen für diverse Plattformen verfügbar. Allerdings befinden sich die Zielplattformen, einmal abgesehen von ERP und MES, ausschließlich innerhalb der Automatisierungsdomäne. Eine breite Anwendung auf Unternehmensebene ist gegenwärtig nicht erkennbar.	1 ↑
Durchdringung	OPC UA hat generell noch nicht die gleiche Verbreitung erreicht wie OPC. Im Kontext des Netz-	0

Kriterium	Begründung	Bewertung
	werk- und System-Management, sowohl innerhalb der Automation wie auch Domänenübergreifend, spielt OPC überhaupt keine Rolle. Wie schon bei der Plattformbindung, ist nicht davon auszugehen, dass OPC UA innerhalb von Unternehmensnetzen und Systemen (MES- und ERP-integration sind Ausnahmen) eine signifikante Verbreitung erlangen wird.	
Mächtigkeit	OPC UA besitzt explizit ein Basisinformationsmodell [132], dazu kommen noch offizielle, und teils ebenfalls standardisierte Erweiterungen. Zusätzliche Erweiterungen sind bereits geplant. Insgesamt sind aber der gegenwärtige und absehbare Umfang und letztlich auch die Ausdrucksstärke in Bezug auf Management-Aspekte, noch nicht umfassend.	1
Erweiterbarkeit	Das OPC UA Informationsmodell ist jederzeit erweiterbar. Einige der bereits vorgestellten Aktivitäten deuten auch darauf hin, dass von der Möglichkeit der Erweiterung Gebrauch gemacht wird.	2
Objekte	Im Rahmen des OPC UA Metamodells, können beliebig aussagekräftige Objekte (Nodes) definiert werden.	2
Beziehungen	OPC UA erlaubt die üblichen objektorientierten Beziehungen	2 ↑
Technologiebindung	OPC UA weist grundsätzlich keine einschränkenden Technologiebindungen auf. Die Kommunikation kann über Webservices erfolgen, die allgemeinhin als flexibel gelten. Stacks sind in verschiedensten Ausprägungen vorhanden und die grundsätzliche Bindung an die Automation spielt an dieser Stelle keine Roll.	2

Kriterium	Begründung	Bewertung
Flexibilität	Durch die konsequente Objektorientierung kann das OPC UA Informationsmodell beliebig erweitert werden um bspw. auch System-Managementaspekte zu repräsentieren. In der Gebäudeautomation [133], [134], [135] findet OPC UA bspw. schon Anwendungen.	2
Integration	Nach der Festlegung in 3.2 (Integration mittels Informationsmodell) ist dieses Kriterium in OPC UA nur mit erheblichen Abstrichen erfüllt. OPC/OPC UA unterstützt diesen Aspekt weitestgehend nicht. Mit der Ausnahme, dass streng genommen das OPC Informationsmodell als Untermenge des OPC UA Informationsmodells betrachten werden kann, bzw. klassische OPC Informationsmodelle mit OPC UA ausgedrückt werden können. Andere Technologien können grundsätzlich integriert werden, explizite Mechanismen sind dafür aber, wenigstens bis zu diesem Zeitpunkt, nicht definiert. Proxylösungen für die schon beschriebene SNMP-Integration sind jedoch möglich.	1 ↑
Einheitlichkeit	Auch wenn die Anzahl der Dienste in OPC UA umfangreicher ist als in anderen Technologien, so ist die Anzahl doch überschaubar und unterliegt keiner Variabilität.	2
Durchgängigkeit	OPC UA macht grundsätzlich keinen Unterschied zwischen realen Objekten. Ein simpler Sensorwert kann genauso abgebildet und gehandhabt werden wie komplexe Auftragsketten die aus einer Vielzahl von Einzelkomponenten bestehen.	2
Standardisierung	OPC UA ist vollständig in Form von IEC Standards definiert.	2

Kriterium	Begründung	Bewertung
Objektauswahl	Die Auswahl der Objekte (Nodes) ist frei und kann zusätzlich noch durch Views und Filter detailliert werden.	2
Modellbereitstellung	Die zentralen Informationsmodelle und deren offizielle Erweiterungen werden von offizieller Seite verwaltet und zur Verfügung gestellt. Herstellererweiterungen können vom jeweiligen Hersteller ebenfalls separat bereitgestellt werden. Ist dies nicht der Fall, kann auch ohne Apriori Kenntnis des Informationsmodells, auf die Daten in einem Server zugegriffen werden. Im View-Service-Set werden entsprechende Dienste zur Verfügung gestellt.	2 ↑
Dynamik	OPC UA erlaubt das dynamische Anlegen und Löschen von Nodes. Da es sich bei OPC UA um eine prototypenbasierte Modellierung handelt, kann man sagen, dass nicht nur Datenelemente, die ein physisches oder logisches Gegenstück haben, angelegt werden können, sondern auch ihre jeweilige Beschreibung in Form von Objektprototypen.	2
Notifikationen	OPC UA bietet umfangreiche und vielseitige Ereignis und Notifikationsmechanismen.	2
Methoden	OPC UA Objekte können beliebig mächtige Methoden definieren.	2 ↑
Transport	Neben einem Binärprotokoll für kritische Anwendungen, bietet OPC eine Webserviceschnittstelle. Während sich die erste noch in der Breite etablieren muss, sind Webservices in Automatisierungsnetzen schon sehr häufig anzutreffen.	2
FCAPS	Praktisch werden durch OPC UA FCP bedient, wobei dies in	(ohne Einfluss)

Kriterium	Begründung	Bewertung
	keinem OPC UA-betreffenden Standard festgeschrieben ist.	
Konzeptkomplexität	Die Komplexität in OPC UA ist generell hoch, nicht unwesentlich durch die strikten Sicherheitsvorkehrungen. Die stetig wachsende Anzahl an Werkzeugen versucht die hohe Grundkomplexität für den Endanwender abzufedern.	(ohne Einfluss)
Anwendungsbereiche	OPC UA wird heute ausschließlich im Automationskontext verwendet. Es ist nicht absehbar, das OPC UA im Unternehmensbereich eine Bedeutung erlangen kann.	(ohne Einfluss)

Im Mittel ergibt sich für die gewerteten Kriterien ein Wert von 1,7 für alle 18 Kriterien, betrachtet man die 5 Kriterien mit überdurchschnittlicher Relevanz gesondert, ergibt sich eine mittlere Bewertung von 1,6. OPC UA eignet sich, sowohl generell wie auch in Bezug auf die überdurchschnittlich relevanten Kriterien, gut bis sehr gut, um den gegenwärtigen und zukünftigen Herausforderungen im Netzwerk- und System-Management der industriellen Automation zu begegnen. Es muss an dieser Stelle jedoch noch einmal deutlich gemacht werden, dass OPC UA sich theoretisch eignet, aber praktisch das System-Management in der Automation zumindest gegenwärtig kein Zielanwendungsgebiet ist. Um für ein durchgängiges Management über Domänengrenzen hinweg in Betracht zu kommen, fehlt es OPC UA zudem am Durchsetzungspotential im Bereich des Systeme-Managements auf Unternehmensebene.

4.5 Die Java Management Extension

Wie schon dem Namen unschwer zu entnehmen ist, steht in diesem Abschnitt ein Ansatz zum System-Management im Fokus, der eng an die Programmiersprache Java gebunden ist. Die erste Version der Java

Management Extension oder kurz JMX wurde im September 2000 freigegeben. JMX war, vor allem am Anfang, mit dem Fokus auf das Management von Anwendungen und der Java Virtual Machine selbst ausgelegt. Im Laufe der Zeit wurde das Anwendungsgebiet erweitert, so dass heute Anwendungen, Endgeräte und Netzwerkhardware gleichermaßen adressiert werden können. JMX ist in seiner gegenwärtigen Form also ein offener Ansatz, mit dessen Hilfe prinzipiell beliebige Ressourcen gemanagt werden können.

Mittlerweile ist JMX Version 1.4 aktuell. Zwischenzeitliche Bemühungen größere Änderungen vorzunehmen und eine Version 2 zu veröffentlichen wurden nach sechs Jahren 2012 vorerst ausgesetzt. Auch in der aktuellen Version sind nicht alle organisatorischen Komponenten die in der Spezifikation [136] erwähnt werden auch spezifiziert. Andererseits sind in der aktuellen Version 1.4 aber auch weitere JSR aufgegangen, etwa JMX Remote API [137], die bislang eine eigene Spezifikation hatten. Im Umfeld von JMX existiert noch eine Anzahl an weiteren Spezifikationen, etwa zur WBEM, TNM oder Webservice Integration, die mittlerweile aber alle zurückgezogen oder suspendiert sind.

4.5.1 Technologische Einordnung

Wie für alle anderen in Abschnitt 4 behandelten Technologien soll auch JMX anhand der vier Management-Modelle technologisch näher eingeordnet werden. Im Fall von JMX ist dies jedoch nicht immer klar möglich, da zum einen in der Spezifikation kein Bezug auf die klassischen Management-Modelle genommen wird, zum anderen die Bindung an Java sehr groß ist. Letzteres sorgt dafür, dass sich die vier Management-Modelle implizit den Java-Sprach- und Systemparadigmen unterordnen. Als vorgreifendes klar erkennbares Beispiel sei das Informationsmodell angeführt. Alle anderen in Abschnitt 4 vorgestellten Ansätze trennen in diesem Belang klar auf, so dass das jeweilige Informationsmodell technologisch unabhängig von Kommunikation und Organisation ist. JMX hingegen beschreibt Informationen ebenfalls direkt in Java-Strukturen.

JMX ist in drei Ebenen *organisiert*. Die unterste Ebene bildet die sogenannte Instrumentierung. Auf dieser Ebene werden reale Ressourcen in das JMX-Datenmodell abgebildet und sind somit für das Java-basierte Management verfügbar. In JMX werden für diese Abbildung von Ressourcen sogenannte MBeans genutzt, die auch als JMX-API bezeichnet werden. MBeans müssen festgelegte Entwurfsmuster umsetzten und, je nach Typ des jeweiligen MBean, festgelegte Interfaces implementieren. Grundsätzlich existieren zwei Typen von MBeans; solche die ihr eigenes Interface implementieren (Standard MBeans) und solche die ein spezielles vordefiniertes Interface implementieren (dynamisch MBeans). Dynamische MBeans bieten dabei den Vorteil der Flexibilität zur Laufzeit. Für Details zu den vorhandenen weiteren (Sub-)Typen sei auf die entsprechenden Stellen in der Spezifikation [136] verweisen. Auf ausgewählte Typen wird im Rahmen der Ausführungen zum JMX-Informationsmodell eingegangen.

Generell gelten für MBeans viele Aussagen die auch für gemanagte Ressourcen in anderen Management-Ansätzen gelten. So müssen alle Belange die Gegenstand des Managements werden soll, durch MBeans repräsentiert werden. MBeans benötigen keinerlei Kenntnis von höheren Schichten der JMX-Organisation. Anzumerken ist, dass bereits die Instrumentation Notifikationen unterstützt. Höhere Organisationsstrukturen in JMX müssen diese Nachrichten lediglich abonnieren.

Die zweite wesentliche Ebene der JMX-Struktur wird durch den sogenannten Agent-Level repräsentiert. Der MBean-Server und die Agent Services sind die wesentlichen funktionalen Komponenten dieser zweiten Schicht. Am MBean-Server müssen alle MBeans registriert werden. Der Sever selbst stellt dann die Sichtbarkeit der MBeans für die Management-Anwendungen her und kontrolliert die eigentlichen Ressourcen, er fungiert also als Informationsbroker und Entkopplung zwischen Management-Anwendung und MBeans. Mit Hilfe der Agent-Services werden unter anderem Überwachungsfunktionalitäten (Monitoring, Timer), dynamisches laden von Klassen und Relations-Dienste angeboten bzw. realisiert. Letztere dienen dabei zum Abbilden

von Assoziationen zwischen MBeans. Neben den genannten Agent-Services, die für jede JMX-konforme Implementierung zwingend erforderlich sind, besteht die Möglichkeit weite Dienste zu definieren und auf Basis von Modulen zu laden. In aller Regel befinden sich MBeans und Agent auf demselben Host, dies jedoch ist keine Forderung der Spezifikation sondern vielmehr gelebte Praxis.

Die dritte und letzte Ebene der JMX Architektur bilden die verteilten Dienste (Service Level). Im Kontext von JMX werden damit Mechanismen bezeichnet, die einem beliebigen Client Zugriff auf Informationen und Dienste auf der Agentenebene ermöglichen. Offiziell ist diese Ebene derzeit nicht Bestandteil der Spezifikation [136] weist darauf explizit hin. Primär wird für den Clientzugriff in der Praxis - Java-typisch - auf RMI vertraut. Generell sind aber auch Adapter für SNMP, WBEM, HTML oder TNM möglich, die jeweils eine protokollspezifische Sicht auf JMX implementieren. Teilweise wurden diese Adapter bereits spezifiziert, gegenwärtig sind alle diese Spezifikationen entweder zurückgezogen oder vorerst stillgelegt.

In JMX bzw. in der dazugehörigen Spezifikation sind, abseits von MBeans und MBeans-Server, keine Aussagen zur Verteilung, Hierarchisierung und Skalierung der einzelnen Komponenten getroffen. Auch die Thematik Agent-Agent Kommunikation wird nicht gesondert behandelt, es wird auf den generellen Funktionsumfang von Java verwiesen.

Das *Kommunikationsmodell* wird in [136] ebenfalls nur implizit beschrieben. Wie bereits erwähnt wird für die Kommunikation zwischen Manager und Agent grundsätzlich auf RMI, mit seinen zugehörigen Spezifikationen, vertraut. Sind etwaige Kommunikationsadapter (z.B. SNMP, WBEM, TNM) im Agent vorhanden, wird entsprechend das Kommunikationsmodell der jeweiligen Technologien genutzt. Ein eigenständiges „Low Level"-Protokoll wie etwa bei SNMP existiert nicht.

Wie auch bei den anderen in Abschnitt 4 evaluierten Ansätzen müssen im Rahmen des Kommunikationsmodells auch die protokollseitig angebotenen Dienste, deren Mächtigkeit und Vielfallt bewertet werden.

Die Darstellung in Tabelle 10 repräsentiert eine grundsätzliche Gruppierung der üblichen JMX-Dienste. Einheitliche Mechanismen, auf Ebene vordefinierter Operationen, existieren für dynamische MBeans, die Anzahl der Operationen (Getter, Setter, Descriptions, uvm.) ist verglichen mit SNMP oder WBEM jedoch erheblich höher. Für Standard MBeans ist die Menge an möglichen Operationen unbegrenzt, da sie beliebige Getter und Setter Implementieren können. JMX bietet letztliche also keinen einheitlichen Pool an Operationen mit denen alle Möglichkeiten der Spezifikation ausgeschöpft werden können.

Die Einordnung von JMX bezüglich des *Funktionsmodells* fällt wieder relativ leicht, denn wenig überraschend ist keines explizit spezifiziert. Dennoch ist es natürlich auch bei JMX so, dass implizit ein Funktionsmodell vorhanden ist bzw. bestimmte Funktionen vom Ansatz selbst erfüllt werden können. In diesem Zusammenhang wird aber auch die

Tabelle 10 Dienste die auf MBeans ausgeführt werden können

Operation / Dienst	Bedeutung
Discovering	Ermöglichen das Auffinden der Dienste und das Identifizieren der durch die MBeans angebotenen Dienstmerkmale.
Reading & Writing	Diese Dienste realisieren das Lesen und Schreiben von Werten/Attributen in MBeans
Performing operations	Mit diesen Diensten werden Operationen ausgeführt die für die jeweilige MBean definiert sind.
Getting notifications	Auf diese Weise können durch MBeans selbst Nachrichten generiert werden, die vom Agent oder direkt vom Manager abonnierbar sind.
Querying	Mit diesen Diensten kann gezielt nach einzelnen MBeans bzw. deren Inhalten gefragt werden. (vgl. Filtering 4.1.1)

Stärke von JMX sichtbar: Auf funktionaler Ebene kann JMX grundsätzlich alles bedienen, was durch die Programmiersprache Java möglich ist. Dennoch ist dieser potentielle Funktionsmodellumfang eben nicht

explizit spezifiziert und so fehlt ihm im weiteren Sinne jede Form der Übertragbarkeit.

Das *Informationsmodell* hinter JMX gestaltet sich in vielerlei Hinsicht strukturell vollkommen anders als in sonst bekannten Management Ansätzen. JMX definiert nicht explizit ein Informationsmodell in einer eigenen Sprache, wie dies von SNMP, OSI oder WBEM bekannt ist. Für die Repräsentation der Informationen wird auf Java-Mechanismen vertraut, genaugenommen auf die Sprache selbst und zusätzlich auf die Anwendung von speziellen Entwurfsmustern. Demzufolge sind die Informationen in Metaklassen, Klassen und Objekten dargestellt und es existieren Vererbung und Assoziation (Relation Service). Eine MBean kann letztlich durch mehrere Klassen implementiert werden. Wie bereits angedeutet wurde, existieren verschiedene Typen von MBeans für unterschiedliche Aufgabenbereiche. Diese werden in der Spezifikation [136] in aller Ausführlichkeit dargestellt. MBean Metaclasses stellen beschreibende Informationen zu den eigentlich MBeans zur Verfügung. Das beinhaltet Informationen bzgl. Attributen, Operationen, Notifikationen und Konstruktoren aber auch über die jeweilige Superklasse.

Die Ausprägung und Struktur der vier Management-Modelle ist in JMX subjektiv und objektiv anders als bei den anderen in dieser Arbeit betrachteten Ansätzen. Grund dafür ist vor allem die sehr starke Bindung an die Paradigmen der Programmiersprache Java in allen Bereichen der Spezifikation.

Nachdem auch JMX kurz technisch eingeführt wurde, wird im folgenden Absatz wieder die Bewertung anhand der aufgestellten Kriterien vorgenommen.

4.5.2 Bewertung

Die Bewertung wird anhand der in Abschnitt 3.2 aufgestellten Tabelle durchgeführt und hat somit direkten Bezug zu Herausforderungen (Abschnitt 2.5), wie man sie heute im System-Management in der Automation wiederfindet. Die jeweils getroffene Bewertung wird kurz begründet, um den Bezug zu den im vorangegangenen Abschnitt

ausgeführten technologischen Eigenheiten zu verdeutlichen. Es wurde im vorangegangenen Abschnitt schon herausgestellt, dass mittels JMX eine System- und Dienste-Management realisiert werden kann, die Ursprünge jedoch im Management der JVM selbst liegen.

Tabelle 11 Bewertung Java Management Extension

Kriterium	Begründung	Bewertung
Plattformbindung	JMX verlangt fast zwingend Java, bzw. eine JVM. Generell ist Java für sehr viele Plattformen zugänglich, für die in der Automation eingesetzten Geräte trifft dies jedoch nicht zu. Die starke Bindung an Java macht JMX für die Automation heute nicht nutzbar.	0 ↑
Durchdringung	Durch die starke Verbreitung von Java ist JMX in vielen Bereichen anzutreffen. In der Automation spielt JMX keine Rolle.	1
Mächtigkeit	Das was man in JMX bzgl. Basisinformationsmodell anbietet ist nur grundsätzlicher Natur, im direkten Vergleich mit WBEM, OSI und SNMP ist die Mächtigkeit des Basisinformationsmodells vernachlässigbar.	0
Erweiterbarkeit	Da hier die vollständige Mächtigkeit von Java zugrunde liegt ist auch das Informationsmodell, im Fall von JMX durch Java-Klassen ausgedrückt, de facto unbegrenzt.	2
Objekte	JMX bzw. schränken den (Funktions-)Umfang von Objekten und Klassen nicht sein.	2
Beziehungen	JMX erlaubt Vererbung und über den Relation-Service auch explizite mit Multiplizitäten versehenen Assoziationen.	2 ↑
Technologiebindung	Auch bei der Technologiebindung wirkt sich die starke Verzahnung mit Java negativ aus.	1

Kriterium	Begründung	Bewertung
	Neben den schon bei der Plattformbindung adressierten Punkten fällt hier z.B. noch ins Gewicht, dass RMI als Standardkommunikationsmittel in der Automation, vor allem auf Feldebene, keine Bedeutung hat. Grundsätzlich ließe sich jedoch auf in der Automation bekannte Mittel zurückgreifen.	
Flexibilität	JMX ist grundsätzlich, und unter Beachtung der schon erwähnten Einschränkungen, für alle Management-Belange einsetzbar.	2
Integration	Integration ist bei JMX nicht zu verwechseln mit der Nutzung von z.B. SNMP als Kommunikationsmedium. Damit ist sozusagen der entgegengesetzte Weg möglich, also die Nutzung von SNMP-Managern zum Management von JMX Agents. Integration in JMX bezieht sich auf die Weiternutzung von vorhanden z.B. Agents und dem entsprechend auch SNMP Informationsmodellen. Dies ist in JMX nicht spezifiziert. Allerdings steht es durch die Mächtigkeit von Java jedem Nutzer frei hier für Integration zu sorgen. Nach der Festlegung in 3.2 (Integration mittels Informationsmodell) erfüllt JMX dieses Kriterium zu einem gewissen Maße, da eben das Informationsmodell selbst auch in Java ausgedrückt wird.	1 ↑
Einheitlichkeit	Für gewisse Gruppen von MBeans (dynamische) kann von Einheitlichkeit bei den Dienstzugriffen gesprochen werden. Allerdings lassen sich letztlich unbegrenzt viele Zugriffsmethoden definieren.	0
Durchgängigkeit	JMX macht grundsätzlich keinen Unterschied zwischen realen Objekten. Jedes reale Objekt oder auch Gruppen realen Objekten können abgebildet und behandelt werden.	2

Kriterium	Begründung	Bewertung
Standardisierung	Für JMX existiert eine Community Spezifikation [136], diese ist selbst allerdings nicht vollständig (Service Level).	1
Objektauswahl	Die Auswahl der Objekte ist frei und kann zusätzlich noch durch Anfragen (Query Dienste) detailliert werden.	2
Modellbereitstellung	Da JMX kein ausgeprägtes Basisinformationsmodell zur Verfügung stellt erübrigt sich ein Teil dieses Kriteriums. Die Informationsstruktur die von einem JMX-Agent verwaltet wird kann jedoch auch zur Laufzeit aus dem „Server" selbst erfragt werden.	1 ↑
Dynamik	JMX lässt hier vollkommen freie Hand, Klassen können zur Laufzeit nachgeladen werden, somit ist der Funktionsumfang der durch einen Agent angebotenen Dienste dynamisch.	2
Notifikationen	JMX bietet beliebig mächtige Notifikationsmechanismen.	2
Methoden	JMX kann für gemanagte Objekte beliebig mächtige Methoden definieren.	2 ↑
Transport	Abseits von RMI als der primäre Kommunikationsmechanismus, ist für JMX kein Transport definiert.	1
FCAPS	Grundsätzlich lassen sich mit JMX alle fünf Teilbereiche abdecken, die Mächtigkeit der Programmiersprache Java, die hier ein wesentlicher Teil des Gesamtkonzeptes ist, bietet alle Möglichkeiten.	(ohne Einfluss)
Konzeptkomplexität	JMX ist, anders als die meisten anderen in dieser Arbeit betrachteten Technologien in nur wenigen Dokumenten mit überschaubarem Umfang definiert, so dass sich ein Einstieg in JMX	(ohne Einfluss)

Kriterium	Begründung	Bewertung
	recht problemlos gestaltet. Hier kann jedoch die Komplexität der Sprache Java nicht außer Acht gelassen werden (die z.B. schon direkt bei der Definition von Informationen benötigt wird), in der Gesamtheit ist JMX eine anspruchsvolle Technologie.	
Anwendungsbereiche	Auch wenn JMX Einzug in einige NMS gefunden hat so wird es doch häufig im Kontext von Java-basierten Anwendungen betrieben.	(ohne Einfluss)

Für JMX ergibt sich im Mittel der gewerteten Kriterien ein Wert von 1,3 für alle 18 Kriterien, betrachtet man die 5 Kriterien mit überdurchschnittlicher Relevanz gesondert, ergibt sich eine mittlere Bewertung von 1,2. JMX eignet sich für den Einsatz zum System-Management in der Automation immer noch knapp überdurchschnittlich. Allerdings muss auch bei JMX wieder einbezogen werden, dass diese Technologie in der Automation nahezu unbekannt ist, was nicht zuletzt daran liegen mag, das gerade Feldgeräte die eine JVM bieten die absolute Ausnahme sind.

Ein weiterer durchaus als negativ zu interpretierender Fakt ist das Nichtvorhandensein eines ausgeprägten Basisinformationsmodells, wie es etwa bei SNMP oder WBEM/CIM der Fall ist. In der Praxis verleitet dies den Anwender, in diesem Fall auch den Gerätehersteller, schnell dazu nicht einheitlich zu arbeiten und so die Übertragbarkeit zu erschweren. Subjektiv ist nicht davon auszugehen, dass JMX in der Automation eine nennenswerte Verbreitung erlangen wird. Auch die Tatsache, dass immer mehr Standard-IT-Technik in der Automation zum Einsatz kommt, vor allem in den höheren Schichten der Automatisierungspyramide, wird daran nichts ändern, denn Feldgeräte werden auch in Zukunft weitgehend auf Java verzichten müssen.

4.6 Web-based Integrated Management Architecture

Anders als die bis zu diesem Punkt betrachteten Ansätze handelt es sich bei WIMA (Web-based Integrated Management Architecture) um einen akademischen Ansatz. Eine prototypische Implementierung der WIMA-Strukturen existiert in Form von JAMAP (JAva MAnagement Platform) [138] zwar, es ist jedoch nichts über den praktischen Einsatz bekannt.

WIMA wurde gegen Ende der 1990er Jahre entwickelt und ist im Wesentlichen in [23] bzw. in der zugrunde liegenden Dissertation [139] mit dem gleichen Titel beschrieben. In dieser Arbeit von Martin-Flatin sind alle ebenfalls zu diesem Thema erschienenen Veröffentlichungen inhaltlich zusammengefasst. Der Zeitraum der Arbeiten an WIMA fällt mit der Zeit zusammen, die man zweifelsfrei als die „Blütezeit" von SNMP bezeichnen kann, aber auch als die Zeit, in der die Kritik an SNMP stärker wurde und diverse Erweiterungen beschrieben wurden, um eben diesen Kritikpunkten zu begegnen. WIMA setzt genau an dieser Stelle an und stellt sich als Management-Ansatz dar, der die Kritikpunkte an SNMP aufgreift und weitestgehend behebt.

Hervorzuheben ist auch der stark integrative Charakter von WIMA, der es erlaubt beliebige Informationsmodelle zu nutzen. Ermöglicht wird dies auch durch eine strikte Trennung von Kommunikations- und Informationsmodell.

Im Rahmen der Arbeiten zu WIMA werden auch Technologien und Ansätze bewertet, die ebenfalls Gegenstand der vorliegenden Arbeit sind, dabei darf nicht außer Acht gelassen werden, dass bspw. WBEM/CIM seit dieser Zeit an vielen Stellen weiterentwickelt worden ist, so dass nicht mehr alle Aussagen, die Martin-Flatin trifft, uneingeschränkt gelten.

Wie schon bei den vorangegangenen Ansätzen, die in dieser Arbeit kurz beschrieben wurden, soll auch WIMA anhand der vier Management-Modelle eingeführt werden. Da in diesem Fall aber eine Einzelarbeit

existiert, werden nur einige überblicksartige Informationen angeführt. An dieser Stelle sei noch einmal auf [23] verwiesen.

4.6.1 Technologische Einordnung

Die Wissenschaftlichkeit in WIMA bringt an dieser Stelle den Vorteil, dass die vier Management-Modelle explizit adressiert werden. Eine Trennung ist dadurch so klar wie in kaum einem anderen zuvor beschriebenen Ansatz möglich. WIMA und der dazugehörige Prototyp basieren durchgehend auf Webtechnologien (XML, http, HTML, Java Servelets etc.), mit allen dazugehörigen Vorteilen.

Martin-Flatin führt in seiner Arbeit klare Gründe an, warum es generell wenig erfolgversprechend ist, mit jedem neuen Problem auch ein neues *Informationsmodell* zu entwickeln. (Diese Ansicht wird auch in der vorliegenden Arbeit geteilt.) Er kommt daher zu dem Schluss, dass keine Notwendigkeit besteht, ein neues Informationsmodell zu entwerfen. Für WIMA lassen sich grundsätzlich alle Informations-Modelle nutzen, die keine zwingenden Abhängigkeiten zu einem der anderen vier Management-Modelle aufweisen. Folglich definiert WIMA selbst kein Informationsmodell, nutzt aber explizit etablierte Informationsmodelle.

Das *Organisationsmodell* wird von Martin-Flatin grundsätzlich neu erarbeitet und vor allem explizit ausgeprägt. Es werden explizit Push- und Pull-Mechanismen beschrieben. Dabei werden, im Gegensatz zu anderen vorgestellten Management-Ansätzen, hauptsächlich Push-Mechanismen für das übliche Management benutzt. Die Datenbehandlung mittels Pull wird ausschließlich für ungeplante (ad hoc) Management-Vorgänge verwendet. Damit wird von Martin-Flatin eine deutliche Reduktion des Kommunikationsoverheads erreicht, da grundsätzlich bei Push-Kommunikation nur dann Netzwerklast erzeugt wird, wenn auch wirklich als relevant definierte Datenänderungen eingetreten sind, die an den Manager übermittelt werden müssen. Martin-Flatin diskutiert die positiven Effekte seines Organisationsmodelles in [140]. Die Begriffe Manager und Agent übernimmt Martin-Flatin bedeutungsgemäß vom OSI-Management bzw. SNMP. Designbedingt wird diese

Struktur allerdings in gewissen Bereichen aufgelöst, in WIMA stellt sich der Manager letztlich als drei-komponentig (Management Station, Management Server und Data Server) dar.

Das WIMA-Organisationsmodell wird in [23] als verteilt und hierarchisch beschrieben. Für die Kommunikation zwischen Manager-Agent und Manager-Manager kommen die gleichen organisatorischen Strukturen ohne Anpassung zum Einsatz.

Das *Kommunikationsmodell* wird für WIMA (aus Sicht des Zeitpunktes der Arbeiten) vollständig neu erarbeitet. Dabei finden sich im Kommunikationsmodell die Push- und Pull-Mechanismen für die Kommunikation zwischen Manager-Agent und Manager-Manager, aus dem Organisationsmodell wieder. Umgesetzt werden beide Kommunikationspfade mittels in http eingebetteter Datenstrukturen, als Transport kommt wie üblich TCP zum Einsatz. Die Datenstrukturen sind dabei vielfältig aufgestellt. Prinzipiell können XML, SNMP aber auch serialisierte Objekte als Datenformat genutzt werden, XML wird jedoch favorisiert. Grundsätzlich können auch Code-Fragmente bzw. Skripte übermittelt werden, die dann vom adressierten Agent ausgeführt werden. Allerdings verfolgt Martin-Flatin diese Idee in seiner Arbeit nur in Ansätzen.

In den Arbeiten zu WIMA wird kein direkter Bezug darauf genommen, welche Management-Operationen zur Verfügung stehen. Es kann nur die Vermutung angestellt werden, dass diese mit dem Informationsmodell zusammenhängen, das gegenwärtig genutzt wird. Beispielsweise sind dies also für SMI/SNMP die in Tabelle 4 aufgeführten Operationen. Organisatorisch steht bei WIMA jedoch die Push-Kommunikation eindeutig im Vordergrund, also das Ereignis/Schwellwert-gesteuerte Übertragen von Informationen vom Agent zum Manager.

In Martin-Flatins Arbeit ist das *Funktionsmodell* sicherlich das, welchem die wenigste Aufmerksamkeit geschenkt wird. Es wird lediglich darauf verwiesen, dass keine Notwendigkeit besteht, ein neues Funktionsmodell zu definieren, welches sich von FCAPS unterscheidet. Dadurch, dass WIMA, bzw. der von Martin-Flatin umgesetzte Prototyp JAMAP,

unabhängig vom Funktionsmodell sind, letztlich jedoch wesentlich auf CIM vertrauen, lassen sich die in Abschnitt 4.3 getroffenen Aussagen unverändert auch auf WIMA übertragen.

4.6.2 Bewertung

Die Bewertung wird anhand der in Abschnitt 3.2 aufgestellten Tabelle durchgeführt und hat somit direkten Bezug zu Herausforderungen (Abschnitt 2.5), wie man sie heute im System-Management in der Automation wiederfindet. Die jeweils getroffene Bewertung wird kurz begründet, um den Bezug zu den im vorangegangenen Abschnitt ausgeführten technologischen Eigenheiten zu verdeutlichen.

Tabelle 12 Bewertung Web-based Integrated Management Architecture

Kriterium	Begründung	Bewertung
Plattformbindung	WIMA ist an keine Plattform gebunden, es wurde sogar explizit mit der Forderung nach Plattformunabhängigkeit entwickelt.	2 ↑
Durchdringung	WIMA bzw. JAMAP wird in der Praxis in dieser Form nicht verwendet, denn im Wesentlichen handelt es sich um einen akademischen Ansatz.	0
Mächtigkeit	Es wird zwar kein Basisinformationsmodell definiert, allerdings war dies eine bewusste Designentscheidung (Wiederverwendung von bestehenden Modellen). WIMA kann jedes beliebige Modell nutzen (darunter CIM und SMI) und übernimmt in dem Fall die Bewertung des Mächtigeren (CIM).	2
Erweiterbarkeit	Hier gilt die gleiche Aussage wie beim Kriterium „Mächtigkeit“. Die Erweiterbarkeit hängt direkt mit dem verwendeten Informationsmodell zusammen.	2

Kriterium	Begründung	Bewertung
Objekte	Auch hier gilt die gleiche Aussage wie beim Kriterium „Mächtigkeit". Die Mächtigkeit MOs hängt letztlich direkt mit dem verwendeten Informationsmodell zusammen.	2
Beziehungen	Hier gilt die gleiche Aussage wie beim Kriterium „Mächtigkeit". Die Fähigkeit zum Herstellen von Beziehungen zwischen Objekten hängt direkt mit dem verwendeten Informationsmodell zusammen.	2 ↑
Technologiebindung	WIMA hat kaum Technologiebindungen (außer beim Transport), sondern empfiehlt lediglich die Nutzung bestimmter Technologien (bspw. XML). Die einzige existierende Implementierung (JAMAP) besitzt einige zwingende Bindungen zu Java (Implementiert in Java, Organisation mittels Java-Servlets etc.). Java ist in der Automation unüblich, es ist nicht abzusehen, dass sich dieser Zustand ändert.	1
Flexibilität	WIMA als Ansatz zum Netzwerk- und System-Management ist, wieder in Abhängigkeit zum eingesetzten Informationsmodell, uneingeschränkt flexibel.	2
Integration	WIMA erlaubt es, mehrere Informationsmodelle verschiedener Management-Ansätze parallel zu nutzen. Integration ist wesentlicher Bestandteil von WIMA.	2 ↑
Einheitlichkeit	Die Frage, in wie weit WIMA den einheitlichen Zugriff auf verschiedene MOs erlaubt kann nicht ohne weiteres beantwortet werden. Werden beispielsweise SNMP/MIB und WBEM/CIM parallel in WIMA genutzt, so erfolgt der Zugriff jeweils mittels der in Abschnitt 4.2.1 und 4.3.1	1

Kriterium	Begründung	Bewertung
	genannten Zugriffsmechanismen, also aus Sicht eines Clients nicht unbedingt einheitlich. Der managerinitiierte Zugriff (in WIMA „pull" genannt) steht jedoch ohnehin im Hintergrund. Für die in WIMA favorisierte „Push"-Kommunikation, in Verbindung mit Servelets und Subskriptionen, spielt Einheitlichkeit eine untergeordnete Rolle. Weiterhin beschreibt Martin-Flatin [140]die bei der „Push"-Kommunikation übermittelten Daten als „selbstbeschreibend".	
Durchgängigkeit	Auch an dieser Stelle kommen die verschiedenen Informationsmodelle, die WIMA nutzen kann, wieder zum Tragen. Letztlich hängt dieses Kriterium direkt vom jeweils eingesetzten Modell ab.	2
Standardisierung	Es ist nicht bekannt, dass abseits der genannten Veröffentlichungen WIMA behandelt wird.	0
Objektauswahl	Auch an dieser Stelle kommen die verschiedenen Informationsmodelle, die WIMA nutzen kann, wieder zum Tragen. Letztlich hängt dieses Kriterium direkt vom jeweils eingesetzten Modell ab. (Objektorientiert vs. Bezeichner-Werte-Paare)	2
Modellbereitstellung	Dieser Punkt steht bei WIMA wiederum nicht im Vordergrund (Push- vs. Pull-Kommunikation). Die schon mehrfach beschriebenen Abhängigkeiten zum genutzten Informationsmodell treffen jedoch auch hier zu.	1 ↑
Dynamik	WIMA beschränkt das dynamische Anlegen von MOs nicht. Auch hier bestehen wieder direkte Abhängigkeiten zu eingesetzten Technologien (SNMP/MIB, WBEM/CIM) bzw. sogar zu deren Implementierung (vgl. 4.3.2)	2

Kriterium	Begründung	Bewertung
Notifikationen	Durch den in WIMA favorisierten „Push"-Ansatz, sind Notifikationen nicht nur möglich, sondern auch so flexibel und leistungsfähig wie in keinem anderen Ansatz.	2
Methoden	Neben den bekannten Querbeziehungen (SNMP, WBEM/-CIM), schlägt Martin-Flatin zusätzlich noch Mechanismen vor, um mittels Mobile Code die Mächtigkeit von Methoden zu steigern.	2 ↑
Transport	WIMA setzte vollständig auf http, was heute auch in der Automation etabliert ist.	2
FCAPS	WIMA kann FCAPS grundsätzlich vollständig abdecken.	(ohne Einfluss)
Konzeptkomplexität	Nimmt man alle Faktoren zusammen, so ist der Anspruch, den WIMA stellt, extrem hoch. Es müssen nicht nur die teilweise selbst schon komplexen Ansätze wie SNMP und WBEM/CIM gemeistert werden, sondern zusätzlich noch die nicht triviale Konfiguration der „Push"-Kommunikation.	(ohne Einfluss)
Anwendungsbereiche	Keine.	(ohne Einfluss)

Im Mittel ergibt sich für die gewerteten Kriterien ein Wert von 1,6 für alle 18 Kriterien, betrachtet man die 5 Kriterien mit überdurchschnittlicher Relevanz gesondert, ergibt sich eine mittlere Bewertung von 1,8. WIMA eignet sich theoretisch überdurchschnittlich gut, in den als besonders relevant eingestuften Kriterien sogar fast uneingeschränkt. Vor allem das in Abschnitt 3.2 kritisch diskutierte Kriterium „Durchdringung" und der ebenfalls schwierig objektiv zu beurteilende Zustand der Standardisierung, die für WIMA jeweils überhaupt nicht gegeben sind, führen letztlich dazu, dass auch im Rahmen dieser Arbeit

WIMA nur grundsätzlich betrachtet wird, nicht aber aktiv eingesetzt werden kann.

WIMA bietet viele interessante Aspekte, etwa die strikte Trennung von Kommunikations- und Informationsmodel, aber auch den Fokus auf die Integration bereits vorhandener Technologien, die eine Betrachtung sinnvoll erscheinen lassen. Hätte WIMA einen gewissen Verbreitungsgrad, wäre es, eben durch den ausgeprägten integrativen Charakter, ein großes Potential für den mittelfristigen Einsatz in der Automation.

4.7 Nicht betrachtete Technologien mit technologischem Bezug zum Netzwerk- und System-Management

In Abschnitt 4 wurde bisher eine ganze Reihe von Technologien zum Netzwerk und System-Management betrachtet. Diese Liste kann keine Vollständigkeit bieten, denn die Anzahl der Technologien, die mehr oder weniger starken Bezug zum Diskursbereich haben, ist sehr lang. Dabei sind Mehrzweckansätze wie etwa Webservices und CORBA [141] genauso vertreten wie Paradigmen wie DEN (Directory Enabled Networking) bis hin zu proprietären Erweiterungen in einigen NMS, die eine gewisse Verbreitung erlangt haben und speziellen Ansätzen um etwa Teilaspekte des FCAPS zu adressieren, beispielsweise RADIUS [142] oder Diameter [143].

Eine Gegenüberstellung verschiedener Management-Ansätze bzw. ihrer Informationsmodelle wird ebenfalls in [144] durchgeführt. Eine detaillierte Wertung kann hier nicht vorgenommen werden. Der Vollständigkeit halber sollen weiterführende Technologien und Ansätze im Folgenden kurz Charakterisiert werden.

Agenten und mobiler Code

Agenten und mobiler Code sind ein weites Feld und eher ein Konzept als eine spezifizierte Technologie. Sie kommen immer wieder im Kontext des Netzwerk- und System-Managements vor z.B. Script MIBs (vgl.

4.2.1). Allerdings fehlt es beiden Konzepten an Durchgängigkeit. In vielen Fällen sind Umsetzungen nur innerhalb eines Unternehmens möglich, da es an weit verbreiteten Standards fehlt, somit sind Laufzeit- oder Ausführungsumgebungen kaum übertragbare Einzellösungen.

Management unter Verwendung von Webservices

Management Ansätze die auf die Nutzung von Webservices zur Kommunikation und Funktionsbeschreibung zurückgreifen existieren in verschiedenen Ausprägungen [145], in Abschnitt 4.3.1 wurde auf Management unter Zuhilfenahme von Webservices im Kontext WBEM/CIM referenziert. Webservices sind letztlich eine allgemein anwendbare Technologie, die nicht grundsätzlich als Management-Paradigma angesehen werden kann.

Management Using CORBA

Hier gilt die Aussage bezüglich der Allgemeingültigkeit in Analogie zu den Webservices. CORBA, als allgemein verwendbare Middleware, kann sicherlich auch zum Management eingesetzt werden. Etablierte Standards, die über generelle Vorgehensbeschreibungen hinausgehen und etwa die vier Management-Modelle referenzieren, existieren nicht.

Frameworx

Entstanden ist Frameworx beim TM Forum [146], ist eine Sammlung von Standards und Best-Practise-Vorgaben zum Management von Diensten innerhalb und zwischen Service Providern. Die Zentrierung auf Business Processes und Service Delivery macht die Verwendung in der Automation schwierig.

Netconf

Auch wenn Netconf als Netzwerk-Management-Protokoll verstanden wird, so wird auch aus der Spezifikation [147] relativ klar ersichtlich, dass der eigentliche Fokus nur ein Teilaspekt des Netzwerk-Managements ist, nämlich das Konfigurations-Management. Für ein ganzheitliches System-Management ist Netconf konzeptionell nicht vorgesehen.

FDT (Field Device Tool)

FDT [148] verfolgt, ganz ähnlich wie Netconf, den Ansatz einer herstellerunabhängigen Konfiguration, dies jedoch speziell im Umfeld der Automation. Für weiterreichende Management-Aufgaben ist FDT nicht konzipiert und wurde deshalb in dieser Arbeit auch nicht näher betrachtet.

PAM (Plant Asset Management)

PAM stellt keine Technologie oder gar ein Protokoll dar, sondern ist eine Bezeichnung für ein Konzept, definiert u.a. in [149]. Auch hier liegt der Fokus nicht auf dem feingranularen Management von einzelnen Komponenten, sondern vielmehr auf (betriebs-)wirtschaftlichen Aspekten von komplexen Installationen.

DEN (Directory enabled Networking)

DEN [150] ist selbst auch keine eigenständige Technologie sondern vielmehr ein Paradigma, das auf CIM aufsetzend, die Möglichkeit bietet Netzwerke bzw. deren Elemente und Dienste darzustellen und auf einen Verzeichnisdienst (z.B. LDAP) abzubilden. DEN geht über das Ziel des System- und Dienste-Managements hinaus, es ist aber nicht ausgeschlossen, dass, sofern Verzeichnisdienste in der Automation eine gewisse Bedeutung erlangen, DEN oder ein ähnlicher Ansatz zukünftig auch in diesem Industriebereich relevant wird.

RADIUS und Diameter

RADIUS [142] und sein Nachfolger Diameter [143] adressieren ebenfalls nur Teilbereiche der klassischen SMFAs (vgl. 4.1), nämlich Accounting und Security. Beiden Technologien dienen dem Authentifikations- und Rechtemanagement im weiteren Sinne.

SDN (Software Defined Networking)

Was SDN [151] ist, aber vor allem was die Grenzen dieses Ansatzes sind, ist kaum scharf abzugrenzen. Grundsätzlich beschreibt SDN die Entkopplung von Daten und Kontrollmechanismen in einem Netzwerk bzw. in der zugehörigen Netzwerkhardware. Auf einer sehr abstrakten

Ebene kann man die Aussage treffen, dass SDN das Netzwerk-Management von einer expliziten Tätigkeit in einem implizite transferiert. Es werden in SDN vielerlei Themen berührt, die auch im Netzwerk-Management von Bedeutung sind (z.B. Routing und Switching), im Sinne dieser Arbeit ist SDN aber kein explizites Management-Paradigma.

Middleware-Lösungen

Middleware im Rahmen der nicht detailliert betrachteten Ansätze zu nennen ist im Grunde falsch. Allen in dieser Arbeit betrachteten Management-Ansätzen ist eines gemein: sie sind streng betrachtet nichts anderes als Middleware-Lösungen. Middleware beschreibt Lösungen die als Mittler zwischen zwei Betrachtungsebenen fungieren – genau das ist es, was auch im Netzwerk- und System-Management adressiert wird, die Abbildung von der Betrachtungsebene „reales Objekt" auf beliebig abstraktes „Managed Object". Weitere konkrete Middleware-Lösungen, etwa aus dem Bereich der Geschäftsprozesse, werden hier nicht diskutiert, ihnen fehlt es in der Regel an direktem Bezug zum Netzwerk- und System-Management-Aspekten.

4.8 Résumé

Die verglichenen Technologien sind, wie zu erwarten war, ganz unterschiedlich für den Einsatz in der industriellen Automation geeignet. Die eine sticht durch ihre Einfachheit und gegenwärtige Verbreitung in der Automation hervor, andere durch gute Erweiterbarkeit, mächtige Modelle oder direkten Bezug zur Automation. In Tabelle 13 sind noch einmal alle Ergebnisse der vorangegangenen Bewertung zusammengefasst und direkt gegenüber gestellt.

Zwei Aussagen lassen sich der Gegenüberstellung auf den ersten Blick entnehmen; es gibt offenbar eine Technologie, die sich ohne wesentliche Abstriche für den heutigen und zukünftigen Einsatz in der Automation eignet WBEM/CIM und es gibt einen klaren Verlierer SNMP. Alle anderen Ansätze haben ihre Vor- aber auch ihre Nachteile, selbst SNMP ist in der Bewertung nicht vollkommen durchgefallen und konnte im Durchschnitt

Tabelle 13 Gegenüberstellung der Netzwerk- und System-Management-Ansätze

Kriterium	OSI	SNMP	WBEM/CIM	OPC UA	JMX	WIMA
Plattformbindung	1 ↑	2 ↑	2 ↑	1 ↑	0 ↑	2 ↑
Durchdringung	0	2	1	0	1	0
Mächtigkeit	1	1	2	1	0	2
Erweiterbarkeit	2	1	2	2	2	2
Objekte	2	0	2	2	2	2
Beziehungen	2 ↑	0 ↑	2 ↑	2 ↑	2 ↑	2 ↑
Technologiebindung	0	2	2	2	1	1
Flexibilität	1	1	2	2	2	2
Integration	1 ↑	1 ↑	2 ↑	1 ↑	1 ↑	2 ↑
Einheitlichkeit	2	2	2	2	0	1
Durchgängigkeit	1	1	2	2	2	2
Standardisierung	2	2	2	2	1	0
Objektauswahl	2	1	2	2	2	2
Modellbereitstellung	1 ↑	0 ↑	2 ↑	2 ↑	1 ↑	1 ↑
Dynamik	2	0	2	2	2	2
Notifikationen	2	1	2	2	2	2
Methoden	2 ↑	0 ↑	2 ↑	2 ↑	2 ↑	2 ↑
Transport	1	2	2	2	1	2
Ø	1,4	1	1,9	1,7	1,3	1,6
ø↑	1,4	0,6	2	1,6	1,2	1,8

aller Kriterien einen neutralen Eindruck hinterlassen. Es sei noch einmal darauf hingewiesen, dass die Gegenüberstellung sich – wie die Kriterien schon verdeutlichen – vorwiegend auf technologische Eigenschaften und

nicht auf „weiche" Fakten wie etwa die Nutzerwahrnehmung, bezieht. Nimmt man zu SNMP – heute der Management-Ansatz in der Automation – und WBEM/CIM – der Ansatz mit dem meisten Potential – noch OPC UA – starke Relevanz in der Automation aber keine Management-anspruch – hinzu ergibt sich eine interessante Konstellation. SNMP konnte sich auch in der Automation bedingt durch seine relativ niedrigen Einstiegsanforderungen (einfaches Protokoll, wenige Zugriffsmethoden, flaches Informationsmodell, etc.) etablieren, wird in der Zukunft aber sicherlich immer mehr an seine Grenzen stoßen. Vor allem vor dem Hintergrund der immer weiter fortschreitenden Integration zwischen Unternehmens-IT, Internet und Automation – Industrie 4.0 – ist das nachvollziehbar. Heute wird im Unternehmensbereich immer mehr auf mächtigere Konzepte, z.B. WBEM/CIM, zurückgegriffen um der gestiegenen Komplexität der Systeme und Dienste sowie den Anforderungen an ihr Management begangenen zu können. Schreitet die Integration der Automation mit Unternehmensnetzen und dem Internet weiter voran, was politisch [152] und wirtschaftlich durchaus angestrebt und gefördert wird, so ist nahezu zwingend davon auszugehen, das auch in den Automatisierungs-Netzen und -Systemen der Zukunft ein Paradigmenwechsel notwendig sein wird.

Aus Sicht der Technologie und vor allem der Automation würde sich hier auch OPC UA anbieten, es ist, wie in 4.4 beschrieben, in der Automation durchaus etabliert, bietet ein hinreichend flexibles Metamodell und die Definition einer Erweiterung für das System-Management ist jederzeit machbar. Ein anderer Umstand wiegt hier jedoch schwerer: es bleibt fragwürdig ob eine Technologie aus dem vergleichsweise kleinen Sektor Automation eine realistische Chance hat, auch in allen übrigen Technologiebereichen Fuß zu fassen. Genau darauf kommt es aber maßgeblich an, wenn man ein durchgängiges Management von möglichst vielen Systemaspekten erreichen will. Das Mapping zwischen verschiedenen Technologien kann, bei einem so grundsätzlichen und infrastrukturell wichtigen Problem, nur eine Übergangslösung sein, die auf Dauer aber durchaus auch Gefahr laufen kann sich beiläufig zu

etablieren – wie es SNMP einst tat. Der technologisch geeignete Ansatz OPC UA scheidet daher also aus strategischen Überlegungen aus.

Aus dem direkten Vergleich zwischen SNMP und WBEM/CIM geht letzteres als in allen technologischen Bereichen überlegen hervor. Wobei der Fokus klar auf dem Informationsmodell CIM als auf WBEM liegt (vgl. 4.3, WBEM ist eine Sammelbezeichnung für eine Reihe an technischen Spezifikationen), was auch aus den Kriterien mit überdurchschnittlicher Relevanz hervorgeht: drei von fünf dieser Kriterien werden maßgeblich durch CIM beeinflusst. WBEM/CIM erscheint also als der Ansatz mit dem meisten Potential, wenn es um die Zukunft des System-Managements in der Automation geht. Am Ende der Ausführungen zu WBEM/CIM in 4.3.2 wurde bereits auf einige, potentiell für die Automation kritische Punkte, hingewiesen dies wird in Abschnitt 5.1 aufgegriffen in dem WBEM/CIM mit SNMP noch einmal auf der Netzwerkebene (Anzahl an Anfragen, erzeugte Netzlast, Vollständigkeit der Ergebnisse, etc.) gegenübergestellt wird.

Uneingeschränkt gültig bleibt die Aussage, dass WBEM/CIM, vor allem CIM, wesentlich komplexer ist und der Einstiegsaufwand, auch für eine ganze Branche größer ist als dies bei SNMP der Fall war und ist. Aus Sicht der wachsenden Ansprüche an Automatisierungsnetze und ihr Management kann dies auf Dauer jedoch kein Argument sein, wachsende Ansprüche bedingen auch im System-Management leistungsfähigere und somit in der Regel aufwändigere Lösungen.

Eine Option für das zukünftige System-Management in der Automation wurde bislang beabsichtigt außen vor gelassen: die vollständige Neuentwicklung eines Management-Paradigmas maßgeschneidert für die Belange der Automation. Es stellt sich hier sofort die Frage, welche Vor- und welche Nachteile eine solche Herangehensweise bringen kann. Zieht man auch dafür noch einmal die aufgestellten Kriterien heran, so wird klar, dass man mit einer vollständigen Neuentwicklung alle Kriterien bis auf die beiden Kriterien Durchdringung und Mächtigkeit sicherlich zur vollsten Zufriedenheit erfüllen kann. Eben diese beiden Kriterien waren es auch schon, denen bei der Aufstellung (vgl. 3.2) eine große Bedeutung beigemessen wurde, die aber aufgrund der

schwierigen Objektivierbarkeit, keines der Kriterien mit überdurchschnittlicher Bedeutung werden konnten. Letztlich sind es aber diese beiden Punkte, die klar gegen die vollständige Neuentwicklung eines Management-Ansatzes sprechen. Bis eine, wie in 3.2 beschrieben, hinreichend große Durchdringung erreicht ist, dauert es in der Regel viele Jahre und dies auch nur unter der Voraussetzung, dass der jeweilige Ansatz akzeptiert wird. Das an dieser Stelle Qualität, Vollständigkeit und nicht zu vernachlässigen auch Zukunftssicherheit nicht unbedingt den Ausschlag geben müssen zeigt SNMP. Für die Reife und Mächtigkeit des Informationsmodells gilt nahezu die gleiche Aussage. Es ist vergleichsweise trivial, wenn auch fragwürdig, ein neues Metamodell zu definieren, dieses Metamodell aber zu nutzen um ein komplexes Informationsmodell zu erstellen ist in der Regel eine Arbeit vieler Jahre. Am Ende würde sich immer noch die Frage stellen, was genau eine Neuentwicklung denn, bezogen auf alle übrigen Kriterien, tatsächlich verbessern könnte, etwa gegenüber WBEM/CIM mit seinen erprobten Technologien und seinem weit gereiften und aussagestarken Informationsmodell, das es sinnvoll erscheinen lässt diesen Weg dennoch zu gehen?

Vor dem schon vielfach erwähnten Hintergrund der fortwährenden Integration zwischen Automation und IT erscheint es vielmehr sinnvoll, den potentesten Vertreter heranzuziehen und diesen um die Belange der Automation zu erweitern wo dies notwendig ist. WBEM/CIM hat sich im Verlauf dieses Abschnittes als geeignet herausgestellt. Deshalb wird dieser Ansatz aufgegriffen und in den folgenden Abschnitten aus dem Blickwinkel der Automation betrachtet und erweiter.

5 WBEM in der industriellen Automation

In den vergangenen Abschnitten wurden verschiedene Ansätze und Technologien, die für das System-Management in der Automation in Frage kommen, ausführlich diskutiert. Es ist wichtig zu betonen, dass eine Umsetzung mit jeder der vorgestellten Technologien prinzipiell möglich ist. Anhand der aufgestellten Kriterien (vgl. Abschnitt 3.2) hat sich WBEM/CIM jedoch als der Ansatz herausgestellt, der gegenüber den anderen möglicherweise technische und organisatorische Vorteile bringt.

In diesem Abschnitt soll folglich WBEM/CIM im Kontext der industriellen Automation betrachtet werden. Neben der Diskussion der Organisationsstrukturen wird in diesem Abschnitt der zweite wesentliche Beitrag der Arbeit, neben der ausführlichen Technologiebewertung, beschrieben: die Erweiterungen am Informationsmodell CIM für aktuelle Belange der Automation.

Alle Betrachtungen und Erweiterungen erfolgen, wie bereits im einleitenden Abschnitt 1.1 ausgeführt, im Fokus Ethernet-basierter Automatisierungsnetze, noch spezieller im Umfeld PROFINET IO. Viele Erwägungen sind ohne weiteres auf andere Ethernet basierte Feldbusse oder sogar auf klassische Feldbusse übertragbar. Besonders für Abschnitt 5.3 gilt: Je abstrakter die Elemente, die dem Informationsmodell hinzugefügt werden, desto direkter lassen sie sich außerhalb des gewählten Fokus wiederverwenden. Was dies im Einzelnen bedeutet und

wie es sich auswirkt, wird in Abschnitt 6 anhand der Anwendungsfälle deutlich.

Im folgenden Abschnitt sollen jedoch zunächst Betrachtungen zur Belastung der Datennetze durch System-Management-Protokolle angestellt werden. Dies wird anhand einer direkten Gegenüberstellung WBEM/CIM gegen SNMP erfolgen, da letzteres in der Automation gegenwärtig de facto Standard ist.

5.1 Leistungsanforderungen im Netzwerk

In den Abschnitten 4.3.2 und 4.8 wurde neben dem Verwerfen der bestehenden Schemata auch die Diskussion um die „lightweight" Fähigkeiten von WBEM/CIM vorenthalten. Dies soll hier nachgeholt werden. Auf Basis der Ergebnisse in [153] kann man zunächst ohne weiteres die Aussage treffen, dass WBEM/CIM, bzw. die auch in der vorliegenden Arbeit favorisierte Kommunikationsform CIM-XML (CIM in XML über http), alles andere ist als effizient und ressourcenschonend. Auch [154] weist ausdrücklich auf Optimierungspotentiale der XML-Encodierung hin. Allerdings vergleicht vor allem [153] in einem sehr engen Rahmen, im Fokus steht dort, wie effizient der jeweilige Management-Ansatz ein und dieselbe Information encodiert und über das Netzwerk kommuniziert. Wie nicht anders zu erwarten, erweisen sich die sehr ausschweifenden XML-Encodierungen hier als relativ aufwendig. Hinzu kommt, dass WBEM/CIM in bestimmten Fällen (z.B. Keys bei „GetInstance") Informationen, die bereits im Request enthalten waren, noch einmal in die Antwort integriert.

Aus Sicht der Anwendung ist es jedoch interessant eine Betrachtung anzustellen, die sich an praktischen Management-Aufgaben orientiert. Hierzu wurden einige kleine Feldversuche durchgeführt. Gegenübergestellt wurden die beiden Operationen „get" (SNMP) und „GetInstance" (WBEM). Beide stellen in der jeweiligen Technologie den Zugriff mit der geringsten Reichweite dar und eignen sich deshalb am besten für den

direkten Vergleich. Die Kommunikation wurde jeweils mit dem Werkzeug Wireshark aufgezeichnet. Auf Seiten WBEM/CIM wurden dabei nur Request und Responds Pakete einbezogen. Etwaige Frames zum Aufbau oder zum Bestätigen der TCP-Verbindung wurden vernachlässigt.

1) Das Lesen eines Objektes (SNMP) bzw. eines Properties (WBEM/CIM). Gelesen wird IF-MIB.ifDescr.1 für SNMP und CIM_IPProtocolEndpoint.Description. Die jeweiligen Antworten sind jeweils nutzdatenbereinigt.

Tabelle 14 Einfache Anfrage SNMP und WBEM/CIM

	Request (Bytes)	Response (Bytes)	Kombiniert (Bytes)	Pro Wert (Bytes)
SNMP	87	87	174	174
WBEM/CIM	1255	332	1587	1587

Während das benötigte Datenvolumen, abgesehen von der Nutzlast, bei SNMP pro Anfrage kaum variabel ist, schwankt es bei WBEM/CIM stark, vor allem was den Request betrifft. Grund hierfür ist, dass im Request sämtliche Schlüsselwerte, die eine Instanz beschreiben, mit übertragen werden müssen. Im Falle dieses Versuches wurde zusätzlich eine einschränkende Liste der in der Antwort zu inkludierenden Properties mit übermittelt, daraus begründet sich auch der kürzere Request in 2)

2) Im zweiten Durchlauf wurden weitere Objekte aus der IF-MIB gelesen (SNMP) sowie alle Properties einer Instanz mittels „GetInstance"(WEBM/CIM). Vergleichend dazu wurden zusätzlich noch alle Instanzen einer Klasse per WBEM/CIM enumeriert. Die Antworten sind auch hier wieder von Nutzdaten bereinigt.

Tabelle 15 Anfrage SNMP und WBEM/CIM im Vergleich

	Request (Bytes)	Response (Bytes)	Kombiniert (Bytes)	Pro Wert (Bytes)
SNMP (5 beliebige)	435	438	873	175
SNMP (BULK) (5 beliebige auf-einanderfolgende)	86	157	243	49
SNMP (BULK) (5 beliebige)	151	157	308	62
WBEM/CIM (eine Instanz) 38 Properties	1179	3716	1587	129
WBEM/CIM (3 Instanzen per EnumInstances) 114 Properties	815	12187	13002	114

Den direkten Vergleich „SNMPget" vs. „GetInstance" kann WBEM/CIM bezogen auf die Properties bzw. Objekte für sich entscheiden, sobald die Anzahl der Properties pro Objekt hoch genug ist. Klar wird, mit den eng gepackten Daten in „SNMPBulk"-Requests, kann WBEM/CIM nicht konkurrieren und erzeugt selbst im günstigsten Fall ca. doppelt so viele Daten. Zu beachten ist jedoch, das SNMPBulk erst ab SNMPv2c vorhanden ist, was in den Automatisierungskomponenten, gegen die getestet wurde, nur in den wenigsten Fällen auch vorhanden war. Auch die Relation von Datendichte zu Informationsgehalt bzw. Informationsdichte ist keinesfalls gegeben.

Optimierungen wären z.B. bereits durch relativ einfache Mittel ohne grundsätzliche Änderungen an der durch die DMTF festgelegten XML-Encodierung zu erreichen, vorausgesetzt natürlich CIMOM und Clients

unterstützen z.B. EXI [155]. Auch der Einsatz von Binärprotokollen ist grundsätzlich möglich. Sollte sich beim Einsatz von WBEM/CIM in der Automation also die Netzwerklast als Problem herausstellen, so könnte ohne weiteres in diesem Bezug optimiert werden. Andererseits gewinnt XML-basierte Kommunikation in der Automation generell an Bedeutung – nicht zuletzt durch OPC UA. Ein grundsätzliches Problem besteht in der Praxis also offenbar gegenwärtig nicht. Auf das Informationsmodell CIM hätte all dies ohnehin keine Auswirkung und wird deshalb in dieser Arbeit auch nicht weiter betrachtet.

In direktem Zusammenhang mit der erzeugten Belastung für das Netzwerk steht die Anzahl an benötigten Anfragen um eine Management-Aufgabe zu lösen. Die Gegenüberstellung SNMP gegen WBEM/CIM in diesem Belang wird Teil der Evaluation (Abschnitt 7) sein, da zunächst die entsprechenden Anwendungsfälle entwickelt werden.

Abseits der Netzwerklast, die durch die Kommunikation der Management-Informationen verursacht wird, nimmt [156] Betrachtungen einzelner WBEM Server (CIM Object Manager – CIMOM) bezüglich ihrer Memory-Footprints vor. Abseits der Ergebnisse, die heute nicht mehr vollumfänglich zutreffen, wird aber untermauert, dass WBEM/CIM bzw. der jeweilige CIMOM durchaus für den Einsatz in performancebeschränkten Embedded Geräten optimiert werden kann.

Der letzte Punkt die Leistungsanforderungen von WBEM/CIM im Netzwerk betreffend ist die Koexistenz mit industriellen Kommunikationsprotokollen bzw. deren Echtzeitanforderungen. Für harte Echtzeitanforderungen (Bsp. PROFINET IRT) gibt es je nach Feldbusprotokoll spezielle Mechanismen um dies zu gewährleisten (Verkehrsplanung etc.). In diesen Bereichen sind per se keine Probleme zu befürchten, da die Echtzeitdaten generell mit Vorrang behandelt werden. Management-Daten haben in der Regel andere, weniger stringente Anforderungen an Echtzeit. Für die Datenkommunikation mit nicht harten Echtzeitanforderungen ergeben sich in der Automation keine geänderten Anforderungen bzgl. Netzwerkbelastung als in der IT. Es ist also erst einmal nicht davon auszugehen, dass hier Probleme entstehen. Selbst mit den vielen in der Praxis notwendigen Workarounds und Proxy-Lösungen, mit deren Hilfe

die Anwendungsfälle umgesetzt und überprüft wurden, konnte zu keiner Zeit, selbst bei aufwendigen Anfragen an den CIMOM, bei den Versuchen und Tests festgestellt werden, dass die reguläre PROFINET IO RT-Kommunikation zu einem Maß beeinflusst wurde, welches einem Versagen des Systems gleichkommt.

5.2 Organisationsstruktur des WBEM für die industrielle Automation

Durch die DMTF sind auf Seiten des Organisationsmodells keinerlei Vorgaben gemacht (vgl. 4.3.1 und [23]). Es gibt aber natürlich Organisationsformen, die sich etabliert haben. Beispiele werden unter anderem in [61] beschrieben. Es obliegt aber letztlich dem Anwender, wie er ein WBEM/CIM basiertes Management-System verteilt.

Die häufigste Organisationsform beinhaltet einen Server (CIMOM) pro physischem System (Server, PC, Netzwerkinfrastruktur etc.). In einem Netzwerk befinden sich in diesem Fall potentiell mehrere WBEM-Server, von denen systemspezifisch gezielt Informationen abgerufen werden können. Das Auffinden von WBEM-Servern im Netzwerk kann über SLP (Service Location Protocol) [157], [81] erfolgen und stellt damit selbst keine Erschwernis dar. Hierarchische Strukturen können umgesetzt werden. Dabei fungiert ein Server auf höheren Strukturebenen als Client und Server zur selben Zeit. Diese hierarchische Strukturierung ist jedoch nicht Bestandteil der Spezifikationen.

Beim eben beschriebenen Herangehen – ein Server pro System – befinden sich die Provider (vgl. 4.3.1) auf dem gleichen System wie ihr zugehöriger Server. Dies ist das gebräuchlichste Verhalten. Darüber hinaus ist es konzeptionell nicht ausgeschlossen, den Provider in gewisser Weise aufzutrennen. Ein Teil des Providers würde dabei nach wie vor auf demselben System laufen wie der CIMOM. Der zweite Teil des Providers kann in diesem Fall auf einem entfernten System, etwa einer performancebeschränkten Automatisierungskomponente, laufen. Wie die Kommunikation zwischen den beiden Teilen des Providers

organisiert ist, obliegt dem Providerentwickler, hier gibt es keinerlei Vorschriften.

Generell schreibt die DMTF – von den zu unterstützenden bzw. zu behandelnden Methoden (z.B. GetInstance) einmal abgesehenen – nicht vor, wie das Interface zwischen CIMOM und Provider gestaltet sein muss. In der Vergangenheit war es so, dass jeder CIMOM sein eigenes Provider-Interface implementierte. Provider wurden für genau diesen CIMOM entwickelt und waren nicht übertragbar. Mit CMPI (Common Manageability Programming Interface) [104] gibt es mittlerweile einen offenen Standard, der es erlaubt, Provider unabhängig vom CIMOM zu entwickeln. Voraussetzung ist lediglich, dass der CIMOM CMPI unterstützt. Für die meisten CIMOM ist dies der Fall (Open Pegasus, ESXi CIM Broker, SFCB, WMI (per Adapter) [158]). CMPI vereinfacht es für Hersteller somit, einzelnen System-Komponenten (Software oder Hardware, aber auch Datenbanken) einem durchgängigen System-Management zugänglich zu machen, ohne mit jedem ihrer Produkte einen eigenen WBEM-Server mitliefern zu müssen. CMPI unterstützt über „Remote CMPI" auch das Verteilen von Providern (Abbildung 9).

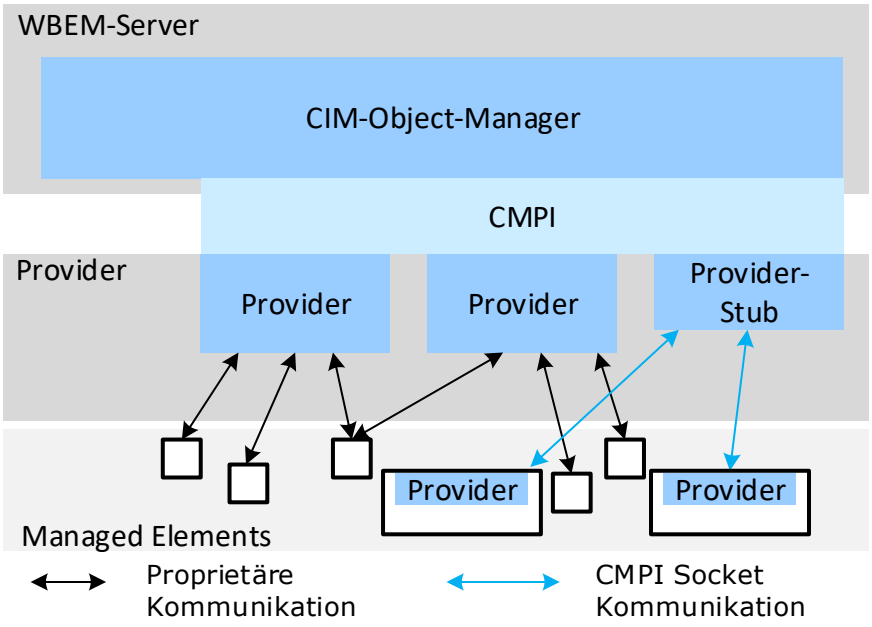

Abbildung 9 CMPI Remote

Wichtig für viele Hersteller, auch in der Automation, ist der Schutz von aufgebautem Know-how. Hier bietet die Auftrennung in offenen Management-Schnittstellen [89] auf der einen Seite und in sich geschlossene Provider auf der anderen die entsprechenden Möglichkeiten. Der Hersteller kann Provider und ggf. Erweiterungen des Informationsmodells entwickeln und die Provider in der Folge binär zur Verfügung stellen. So sind seine Produkte einem System-Management zugänglich, ohne dass die Notwendigkeit besteht alle Schnittstelleninformationen publik zu machen. In der Windows-Welt ist es beispielsweise üblich, die Integration in WMI mit der Installation der Software automatisch durchzuführen, sofern dies vom Hersteller vorgesehen und vom Anwender gewünscht ist. Für Anwendungssoftware oder auch Gerätetreiber gilt dies in gleichem Maße.

Für die Automation können, je nach Anwendungsszenario, beide Ansätze (lokale Provider und verteilte Provider) genutzt werden. Der Betrieb von WBEM-Servern auf leistungsstärkeren Automatisierungskomponenten wie etwa Speicherprogrammierbaren Steuerungen (SPS) ist nicht abwegig, sofern die jeweiligen Hersteller hier die Möglichkeiten vorsehen. Für leistungsschwache Komponenten, die in der Regel nur in Zusammenarbeit mit leistungsstärkeren betrieben werden können, bietet sich das Verteilen der Provider an. Auch in diesem Fall ist die Mitwirkung der Gerätehersteller notwendig.

Dass es bereits WBEM-Server am Markt gibt, die sich für den Betrieb in Embedded-Umgebungen eignen, wurde in [156] nachgewiesen. Aktuell ist jedoch nicht bekannt, dass Embedded-Automatisierungskomponenten (SPS, remote IOs) existieren, die einen CIMOM integrieren oder die Installation eines solchen ermöglichen. An dieser Stelle können nur die Hersteller selbst tätig werden.

Gegenwärtig, vor allem um einen zeitnahen Einstieg zu ermöglichen, wird eine dritte Organisationsform benötigt. Dabei werden aus WBEM-Sicht die Provider lokal auf leistungsstarker Hardware (PC) betrieben. Die Provider selbst interagieren mit den „Managed Objects" jedoch über das Netzwerk. Zurückgegriffen wird dabei auf verschiedenste Protokolle.

Abhängig vom konkreten Anwendungsfall (siehe Abschnitt 6) können das Protokolle der Automation (z.B. PROFINET IO), aber auch Terminal Protokolle (z.B. Telnet) oder andere Management-Protokolle (SNMP) sein. Obwohl dadurch ein gewisser Mehraufwand notwendig wird, schließlich muss eine ganze Reihe unterschiedlichster Protokolle gehandhabt werden, und auch die Belastung für die Kommunikationspfade potentiell steigt, ist dies gegenwärtig der einzige Weg, um ein Management mittels WBEM/CIM in der Automation umzusetzen. Auf die Anpassungen und Erweiterungen am Informationsmodell hat der Einsatz dieses Proxykonzeptes keinen Einfluss, es ist, einmal abgesehen von CIMOM Interna, ohnehin unabhängig von der Organisationsform.

5.3 Common Information Model für die Automation

Nach den umfangreichen Betrachtungen zur generellen Eignung verschiedener Management-Ansätze und ihrer Informationsmodelle wurde WBEM/CIM (vgl. 4.3) als angestrebte Zieltechnologie identifiziert. Obwohl CIM (vgl. 4.3.1) ein durchaus umfangreiches und ausdrucksstarkes Modell darstellt, muss es für die Belange der Automation ausgeprägt werden. Teilweise müssen Technologien, die bislang nicht in CIM abgebildet sind, integriert werden (bspw. Feldbusprotokolle, Dienste zur Identifikation), teilweise müssen Sichtweisen auf (Teil-)Systeme so dargestellt werden, dass sie mit den Begriffen und Sichtweisen der Automation vereinbar sind. Einer der wichtigsten und umfangreichsten Arbeitsschritte bei der Erweiterung des Informationsmodells ist jedoch seine Erschließung. Dieser Aufwand kann umgangen werden, indem nur das Metamodell [92] verwendet wird, nicht jedoch Core- und Common-Schemata. Es sind also zwei unterschiedliche Vorgehen möglich:

(i) Neuentwicklung eines Informationsmodells

(ii) Nutzung und Erweiterung der vorhandenen Informationsmodellelemente

Beide Lösungsansätze bringen ihre ganz eigenen Vor- aber auch Nachteile mit sich. In den folgenden beiden Abschnitten werden sie kurz bewertet und der geeignetere, den Anforderungen der Automation entsprechend, verfolgt.

5.3.1 Neuentwicklung eines Informationsmodells für die Belange der Automation

CIM räumt seinen Nutzern die Möglichkeit ein, nicht das von der DMTF betreute Informationsmodell (dieser Arbeit zugrundeliegend CIM 2.36.0) zu nutzen und zu erweitern, sondern unter Nutzung des Metaschemas ein vollkommen neues Informationsmodell zu entwickeln. Obwohl dies ein vollkommen valider Ansatz ist, so muss doch in Betracht gezogen werden, dass durch ein solches Vorgehen viele der Vorteile, die das umfangreife Informationsmodell der DMTF mitbringt, verloren gehen würden.

Die Mächtigkeit von CIM basiert in weiten Teilen auch auf einem hohen Maß an Wiederverwendung von definierten Informationsklassen in immer neuen Anwendungen. Eine Neudefinition von bereits modellierten Informationen würde dem widersprechen. Durch das Verwerfen von CIM Core und Common würde für viele Aspekte jedoch genau diese Art von Neudefinition, eigentlich bereits modellierter Informationen, notwendig. Als Beispiel kann hier ganz allgemein die Modellierung von Ethernet herangezogen werden. Diese Technologie kommt in der Automation im Wesentlichen unverändert gegenüber den in CIM modellierten Informationen zum Einsatz. Eine Neumodellierung würde nicht nur der Wiederverwendung entgegenstehen, sondern, da man davon ausgehen muss, dass nicht vollkommen identisch modelliert wird, auch den eigentlich erwünschten und angestrebten Integrationsbemühungen zwischen beiden Domänen, da an dieser Stelle aufwendige Abbildungen (Mappings) zwischen beiden Modellen notwendig wären. An dieser Stelle könnte die Betrachtung der Herangehensweise „Neuentwicklung eines Informationsmodelles“ bereits abgebrochen werden, denn damit würde den eigentlichen Zielen dieser Arbeit entgegengewirkt.

Die DMTF definiert mit Profilen (vgl. 4.3.1) quasi Implementierungstemplates für Standardmanagement-Aufgaben. Profile beschreiben, welche Klassen instanziiert werden müssen/sollten, um diese Aufgabe mittels CIM zu lösen. Dieses Vorgehen ermöglicht nicht nur einen schnellen Einstieg in CIM, sofern die jeweilige Management-Aufgabe hinreichend genau abgedeckt wird, sondern stellt auch sicher, dass Clients generisch entwickelt werden können. Auch dieser Vorteil würde mit der Neuentwicklung eines Informationsmodells, zumindest vorerst, verloren gehen, da die Kompatibilität zwischen DMTF Profilen und neuem Informationsmodell wahrscheinlich nicht gegeben wäre.

Auf subjektiver Ebene ist als kritischer Punkt bei der Neuentwicklung von Modellen, die als Ziel die Abbildung ganzer Technologiedomänen haben, noch der schwierige Reifungsprozess des Modells selbst zu nennen. Modelle durchleben in der Regel viele Iterationsstufen von den ersten Entwürfen bis zu einer markttauglichen Reife. Fehlt der dabei permanent stattfindende Überprüfungs- und Verbesserungsaspekt, ist zum einen davon auszugehen, dass fehlerbehaftete Modelle entstehen und zum anderen Fehler erst so spät erkannt werden, dass mit dem Beheben evtl. Kompatibilitätsprobleme mit bereits im Einsatz befindlichen Versionen entstehen. Da bei einer Modellneuentwicklung Informationen neu abgebildet werden müssten kann selbst hier davon ausgegangen werden, dass Fehlerfreiheit nicht gewährleistet werden kann.

Die Neuentwicklung eines Informationsmodells im Rahmen dieser Arbeit hätte jedoch nicht nur negative Aspekte. Vor allem die im Fall einer Neuentwicklung geringe bzw. nicht vorhandene Komplexität von bereits bestehenden Modellkomponenten würde den Einstieg vereinfachen. Dies gilt sowohl für die Erstellung/Erweiterung des Modells wie auch für die Anwendung durch Dritte. Es müsste nicht für jede Information – mit Relevanz für die Automation – zunächst überprüft werden, ob diese bereits abgebildet ist bzw. die Position im bestehenden Modell identifiziert werden, die sich am ehesten eignet um die jeweilige Information, in Form einer neuen Klasse, zu modellieren. Generell ist davon auszugehen, dass ein speziell für die Automation entwickeltes Informationsmodell auf Basis des CIM Metaschemas einen geringeren Umfang hätte,

als alle CIM-Schemata zusammen genommen. Wieviel geringer der Umfang allerdings ausfallen würde, lässt sich kaum abschätzen.

Zwei Vorteile, die mit einer Neuerstellung ohne Zweifel gewonnen werden können, betreffen die Notwendigkeit von Kompromissen und das Entfernen von Altlasten. Zum Beispiel: Die exakte Bedeutung der Begriffe Geräte, Devices, Systeme und Module ist zwischen CIM und der allgemeinen Automatisierungssichtweise nicht deckungsgleich. Altlasten sind ein Problem, das CIM, wie jedes Modell, das über Jahre wächst und immer wieder neuen technologischen Gegebenheiten angepasst wird, hat. Neben dem Qualifier „Depricated" [92], der in CIM diesen Sachverhalt unmittelbar zum Ausdruck bringt, existieren auch Modellelemente [92] die aus Gründen der Kompatibilität nicht auf „Deprecated" gesetzt werden, da noch viele Implementierungen existieren, die den jeweiligen Mechanismus noch benutzen, so z.B. die Klasse „CIM_StatisticalInformation". Diese Fragmente aus früheren Modellversionen wären bei einer Neuentwicklung kein Problem mehr, was der generell hohen Komplexität der Informationsmodelle sehr entgegenkommen würde.

Letztlich wiegen die Nachteile, die eine Neuentwicklung mit sich bringen würde, jedoch so schwer, dass dieser Ansatz nicht verfolgt wird.

5.3.2 Nutzung und Erweiterung des Common Information Models für die Automation

Ein wesentlicher Beitrag der Arbeit wurde schon herausgearbeitet: die Identifikation von Ansatzpunkten im Common Information Model für die Belange der industriellen Automation. Folglich kann es, wie im vorangegangenen Abschnitt diskutiert, nicht sinnvoll sein, mit „möglichst vielen neuen Klassen" CIM für die Automation zu erschließen oder gar ein vollkommen neues Modell zu entwickeln. Dieses Vorgehen würde zwar den Einstieg erleichtern, Kompromisse vermeiden und die Quantität der Neuentwicklungen erhöhen, stünde aber in weiten Teilen im Konflikt mit dem Ziel der möglichst fließenden Integration zwischen Automation und IT.

Es stellt sich nun die Frage: Wie lässt sich die Automation – generell und speziell im Kontext dieser Arbeit – in CIM abbilden? Die Ansatzpunkte dabei sind vielfältig. Die folgende Aufstellung soll einen groben Überblick geben, welche in CIM existierenden Klassen evtl. Superklassen für bestimmte Automatisierungsthemen sein können, bzw. in welchem der Common Schemata sich wesentlichen Klassen der jeweiligen Management-Aufgabe befinden. Querbeziehungen (Assoziationen), Attribute (Properties) oder auch weitere Abstrahierungen spielen dabei zunächst keine Rolle, sie sind Gegenstand der speziellen Anwendungsfälle in 6. Für die folgenden Ausführungen wird jeweils das CIM Schema in der Version 2.36.0 [159] herangezogen. Es wird darauf verzichtet, Klassen, deren Namensschema mit „CIM_" beginnt, im Einzelnen zu referenzieren (etwa durch explizite Nennung eines definierenden Dokuments). Klassen werden durch Anführungszeichen umschlossen, Schemata werden ohne diese genannt.

(i) Abbildungen von Protokollen und Protokollfamilien (z.B. PROFINET IO – direkt aufsetzend auf Ethernet, über UPD/IP, aber auch DCP) lassen sich „CIM_ProtocolEndpoint" unterordnen. Etwaige mit dem jeweiligen Protokoll in direktem Zusammenhang stehende, managebare Dienste lassen sich als „CIM_ProtocolService" abbilden und sind somit ebenfalls Bestandteil des Common Schemas CIM_Network.

(ii) Leistungsanalysen, im Sinne von Performance (FCAPS), lassen sich in das Common Schema CIM_Metrics sowie als Spezialisierung der Klasse „CIM_StatisticalData" in CIM einordnen. Das auf Teilen dieser Arbeit basierende Projekt SMartA [160] nutzt diese Strukturen, um mehrstufige Metriken zur Bewertung der Verbindungsgüte zwischen (industriellen) Kommunikationspartnern abzubilden.

(iii) Alle Belange rund um Software, egal ob Firmware, Betriebssystem, oder Anwendersoftware, lassen sich in CIM im Wesentlichen drei Klassen unterordnen –

„CIM_SoftwareIdentity“ (CIM_Core Schema), „CIM_SoftwareFeature“ und „CIM_SoftwareElement“ (CIM_Application Schema). Die Zuordnung ist abhängig davon, welcher Aspekt jeweils im Vordergrund steht. Des Weiteren werden über Assoziationen von den genannten Klassen Aspekte wie etwa die Installation von Software, die Zuordnung zu Hardware und Fähigkeiten (Capabilities) abgebildet.

Die Granularität, in der Software durch die oben genannten Klassen abgebildet werden kann, ist grundsätzlich beliebig. Dies ermöglicht im Kontext der Automation z.B. auch die Abbildung bzw. das Steuern (Start, Stop, Status) von SPS-Programmen in CIM. Über die Fähigkeiten zur Softwareinstallation, den damit in Verbindung stehenden „Services“ bspw. zum Ausführen, könnte das System-Management-Aufgaben, die im Allgemeinen den MES zugeordnet werden, übernehmen. Als Beispiel ist hier die rezeptspezifische Auswahl von PLC-Programmen zu erwähnen. Vollkommen unerwartet ist dieser Ansatz nicht, da sowohl das System-Management wie auch MES als Middleware betrachtet werden können. Die Grenzen zwischen System-Management und Prozesssteuerung würden bei einem solchen Vorgehen jedoch verschwimmen. Welche positiven aber auch negativen Effekte das Integrieren dieser beiden Domänen mit sich bringen würde, wird im Rahmen der vorliegenden Arbeit jedoch nicht vertieft.

(iv) Alarme, Meldungen oder allgemeiner Zustandshistorien können als Spezialisierung von „CIM_Log“ dargestellt werden. „CIM_Log“ ist Bestandteil des Common Schemas CIM_System (siehe (viii)).

(v) Meldungen mit der Anforderung an unmittelbare Kenntnisnahme, können als Spezialisierung von „CIM_Indication“ abgebildet werden. „CIM_Indication“ ist Teil des Common Schemas CIM_Event. Die Klasse „CIM_Indication“ würde in der Anwendung nicht unbedingt genutzt werden, um direkt

von ihr zu spezialisieren, vielmehr wird von „CIM_ClassIndication“ und „CIM_InstIndication“ für Informationsmodell-betreffende Ereignisse sowie von „CIM_ProcessIndication“ für den realen Prozess betreffende Ereignisse, abgeleitet.

(vi) Physische Eigenschaften von industriellen Installationen – Montagepositionen in Racks bzw. auf Hutschienen, geographische Lokation der Installation, Gehäuseabmessung etc. – können als Spezialisierung von „CIM_PhysicalElement“ bzw. einer den jeweiligen Anforderungen entsprechenden Unterklasse abgebildet werden. „CIM_PhysicalElement“ selbst ist Teil des Core Schemas, Unterklassen gehören dem CIM_Physical Schema an.

(vii) Daten zur Identifikation von Geräten (z.B. PROFINET IO I&M oder EthernetIP bzw. CIP Identity Object) gehören zu den Informationen, die sich nicht auf den ersten Blick in CIM einordnen lassen. Identifikationsinformationen können als teils statische, teils variable Konfigurationen angesehen werden. Informationen der Art Konfiguration können als „CIM_SettingData“ (Core Schema) abgebildet werden. An dieser Stelle müssen jedoch umfangreichere Modellierungsarbeiten durchgeführt werden. Zum einen sollen auf abstrakter Ebene alle Identifikationsinformationen verschiedener Feldbusse gleich behandelt werden, zum anderen haben die einzelnen Feldbusorganisationen auf spezieller Ebene natürlich unterschiedliche Informationen vorgesehen – teilweise auch nur durch Umbenennung. Gerade für die Abbildung von Identifikationsinformationen auf abstrakter Ebene ist in der Zukunft eine Standardisierung seitens der DMTF und der Feldbusorganisationen notwendig.

(viii) Geräte und deren Komponenten bzw. das sie betreffende Management sind die zentralen Elemente. Automatisierungstechnische Geräte lassen sich aus Sicht des CIM auf drei Arten beschreiben:

a. nach ihren physischen Eigenschaften (vgl. (vi))
b. nach ihrer logischen Struktur
c. als funktionale Einheit

   Für die Beschreibung nach (viii)b eignet sich „CIM_LogicalDevice". Durch „CIM_LogicalDevice" können sowohl Geräte wie auch ihre Komponenten (steckbare Module, Busanschlüsse etc.) beschrieben werden. Logische Geräte können, müssen jedoch selbst nicht eigenständig betreibbar sein. Durch „CIM_LogicalDevice" sind alle Eigenschaften zu beschreiben, die Gegenstand des Managements der jeweiligen logischen Komponente sind oder sein sollen. Beispiel: die aktuelle –logische – Position, an der ein Modul auf einer RemotIO-Kopfstation gesteckt ist.

   Für die Komposition einer Anzahl solcher logischen Geräte zu einer funktionalen Einheit ((viii)c) wird „CIM_ComputerSystem" genutzt, wobei der „CIM_ComputerSystem" nicht nur eine simple Aggregation von Teilen darstellt, sondern diesen Teilen erst eine Gesamtfunktion gibt. Im Gegensatz zu „CIM_LogicalDevice" hat jedes „CIM_ComputerSystem" sowohl im Modell wie auch in der Realwelt eine eindeutige Identifikation. „CIM_LogicalDevice" ist Bestandteil von CIM_Core, „CIM_ComputerSystem" ist bereits Bestandteil von CIM_System. Die Konzepte zu „CIM_ComputerSystem" und „CIM_LogicalDevice" sind so grundlegend für die Anwendungsfälle in 6, dass sie in 6.1 im Detail eingeführt und entsprechend für die Automation erweitert werden.

Obwohl die obenstehende Zuordnung nur konzeptioneller Natur ist, zeigt sie doch, dass nicht „die" Stelle in CIM ermittelt werden kann, an der sich die Belange der Automation einordnen lassen, vielmehr sind die Ansatzpunkt weit verteilt und abhängig vom jeweiligen Anwendungsfall bzw. von der umzusetzenden Management-Aufgabe. Auch ließen sich noch weitere Themen innerhalb der Automation finden, die Gegenstand eines System-Managements sein könnten, etwa die Überwachung von

Batchprozessen, das Condition Monitoring oder das Life Cycle Management [101]. Hinzu kommen Management-Aufgaben, die in der Automation im Wesentlichen identisch sind zu denen in der IT – also all das, was z.B. das Medium Ethernet an sich betrifft (oder IP, UDP, etc.) oder das Management der auch heute schon im Feld vorhandenen IT-Systeme.

Es sind für die Automation natürlich auch noch andere als die hier direkt erwähnten Common Schemata relevant, z.B. CIM_Network, die verschiedene Teilaspekte aus anderen Schemata aufgreifen und neu in Relation setzen. CIM_Network bezieht etwa Aspekte aus „CIM_System" (Superklasse zu „CIM_ComputerSystem"), „CIM_LogicalDevice", „CIM_-ServiceAccessPoint" (Superklasse zu „CIM_ProtocolEndpoint) mit ein, um das Netzwerk als solche in CIM zu repräsentieren.

Nach den obenstehenden konzeptionellen Zuordnungen wird klar, dass sich die meisten – wahrscheinlich sogar alle – Belange der Automation in CIM abbilden lassen. Einzig die Position dieser Abbildung ist nicht immer einfach zu identifizieren. Der Anpassungsaufwand, vor allem in Form von Spezialisierungen, hängt davon ab, ob in der IT – somit in CIM – ein vergleichbarer Anwendungsfall bzw. eine vergleichbare Management-Aufgabe existiert.

Die Integration von Informationen aus bestehenden – in der Automation in der Regel SNMP basierten – Management-Lösungen auf Basis des Informationsmodells („MIB nach CIM") konnte bereits in 3.2 als überdurchschnitt relevantes Kriterium identifiziert werden. In 4.3.2 wurden kurz einige Informationen, das Verfahren der Integration in CIM betreffen, gegeben. Die Abbildung von bestehenden (automatisierungsspezifischen) MIBs nach CIM erfolgt mittels des Qualifiers „MappingString". Es wird jedoch explizit nur eine Abbildung von SMIv2 Object [78] nach CIM Property [92] vorgenommen. Für die Abbildung von bestehenden MIBs nach CIM muss also zunächst wieder die geeignete Klasse in CIM identifiziert werden, die das Property besitzt. Dabei gilt es jedoch zu beachten, dass die Informationen/Daten in SMIv2 vollkommen anders strukturiert sind, so das Informationen die in SNMP

direkt „benachbart“ (aufeinander folgende Object Identifier (OID)) sind in CIM nicht zwangsläufig Properties derselben Klasse sein müssen.

6 Realisierung ausgewählter Anwendungsfälle mittels WBEM

In Abschnitt 5.3.2 wurden ganz grundsätzliche Entscheidungen, die Modellierung von automatisierungstechnischen Aspekten in CIM betreffend, vorgestellt. Nun sollen anhand der in Abschnitt 2 aufgestellten Anwendungsfälle, detaillierte Erweiterungen (Extensions) des Informationsmodells entwickelt werden. Es wird dabei immer wieder notwendig sein, das abstrakte Niveau zu verlassen und Modellierungsentscheidungen zu treffen, die feldbus- oder herstellerspezifisch sind. Konkret bedeutet das wie in Abschnitt 1.2 schon angedeutet, die Modellierung erfolgt speziell für die Industrial Ethernet Technologie PROFINET IO [161], [162]. Teilweise sind sogar noch weitere Spezialisierungen notwendig, vor allem immer dann, wenn Gerätespezifika benötigt werden um einen Anwendungsfall umzusetzen. In diesem Fall wird die Modellierung sogar speziell für Profinet IO Geräte der Firma Siemens durchgeführt. In den neu zu CIM hinzugefügten Klassen werden sich die genannten Spezialisierungen immer dadurch manifestieren, dass etwaige Klassen „Profinet" oder „Siemens" im Namen tragen.

Um in Abschnitt 7 die Fähigkeiten der Modellerweiterungen evaluieren zu können, ist es notwendig das Modell zu instanziieren. Dies erfolgt in WBEM/CIM über sogenannte Provider. Provider haben jedoch nicht nur die Aufgabe, einfache Werte aus den realen Managed Objects in das Modell zu überführen, sondern können bidirektional agieren und entsprechend Werte in die realen Objekte schreiben. Das Ausführen von in CIM-Klassen definierten Methoden auf einem realen Objekt liegt ebenfalls im Verantwortungsbereich des Providers. Die Providerentwicklung wird durch das CMPI [104] erleichtert. Weiterhin existieren

Frameworks wie KonkretCMPI [163] und SimpleWBEM [164], die direkt bei der Generierung von Providerquellen unterstützen. Für die Instanziierung des Modells im Rahmen dieser Arbeit wurde SimpleWBEM genutzt.

Der Namensraum für alle im Rahmen der vorliegenden Arbeit hinzugefügten Klassen ist „TUDIC_". Klassen, die sich lediglich in „TUDIC_" und „CIM_" unterscheiden, sonst aber einen identischen Klassennamen aufweisen, deuten in aller Regel darauf hin, dass mit der „TUDIC_" keinerlei neue Properties hinzugefügt wurden. Diese Klassen haben lediglich einen strukturierenden Charakter bzw. werden benötigt um einen Provider für sie zu registrieren.

6.1 Das Gerät als zentrale Komponente für das Management

Die Geräte (Steuerungen, RemoteIOs, Infrastrukturgeräte) sind für diese Arbeit von zentraler Bedeutung. So zentral, dass es im Hinblick auf die Anwendungsfälle sinnvoll erscheint, die Modellierungsentscheidungen, die Geräte aus Sicht ihrer Komponenten und auch als Gesamtsystem betreffend, vorangestellt zu betrachten. Die systematische Herangehensweise (vgl. 5.3.2(viii)) wird in Abbildung 10 dargestellt.

Es soll hier nicht im Einzelnen auf die Bedeutung und evtl. Abstammung jeder Property [92] eingegangen werden. Werden bestimmte Properties, z.B. bei der Realisierung der Anwendungsfälle, benötigt, wird dies nachgeholt. Gleiches gilt auch für Qualifier [92], mit denen Klassen, Properties oder auch Methoden annotiert sind.

„TUDIC_Device" stellt die Oberklasse für alle automatisierungstechnischen Geräte als Gesamtsystem dar. Die jeweiligen Spezialisierungen bilden zum einen Besonderheiten der Feldbusse ab und dienen zum anderen als Verbindung zwischen Modell und realem Gerät über einen - in diesem Fall feldbusspezifischen – Provider. „TUDIC_Device" stellt grundsätzlich jedes projektierte Gerät dar, über ererbte

Properties kann die in der Automation bekannte Life-List aller erreichbaren Geräte ausgedrückt werden.

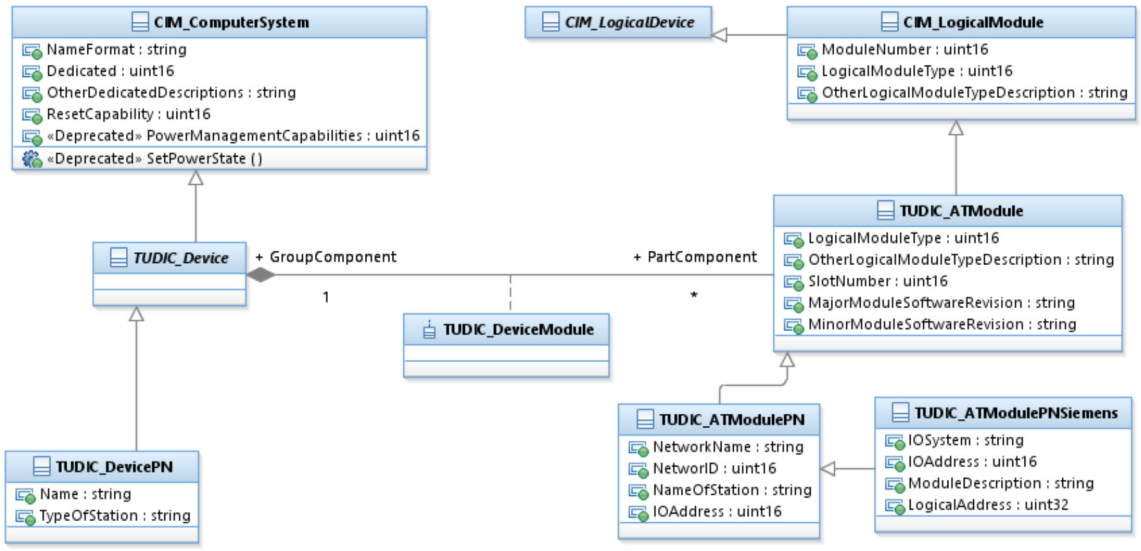

Abbildung 10 TUDIC_Device und TUDIC_LogicalModule

Für ein Profinet IO-Gerät wird eine Klasse, hier „TUDIC_DevicePN", beispielsweise die Property „CIM_System.Name", die sie ererbt hat, lediglich in „TUDIC_DevicePN.NameOfStation" umbenennen. Den überwiegenden Teil an Properties ererbt „TUDIC_DevicePN". Damit wird Profinet IO-spezifisch der Name des jeweiligen Gerätes als Identifikation genutzt. Weitere Merkmale, anhand derer eine Geräteerkennung festgemacht werden kann, bspw. MAC oder IP, sind in dieser Darstellung nicht enthalten, da sie durch entsprechende Strukturen in den Spezialisierungen zu „CIM_ProtocolEndpoint" abgebildet sind.

Durch das Instanziieren geeigneter Assoziationen, etwa „TUDIC_DeviceModule", wird die Verbindung zu den logischen Komponenten des Systems „Device" hergestellt. Jedes System („TUDIC_Device") setzt sich aus einer Anzahl logischer Geräte zusammen. Da Geräte in der Automation häufig modular aufgebaut sind, wurde „TUDIC_ATModule" als Spezialisierung von „CIM_LogicalModule" eingeführt. Jedes Gerät im Sinne von „TUDIC_Device" besteht also aus mindestens einem Modul, dies geschieht in Anlehnung an [165]. „TUDIC_ATModule" ist dabei selbst wieder die Oberklasse für weitere Detaillierungen bzgl. Feldbus und noch spezieller bzgl. Hersteller. Letztere ist notwendig, da Hersteller

teilweise noch zusätzliche Informationen zur Identifikation der Geräte(-komponenten) heranziehen, bzw. Informationen aus dem Engineering benötigt werden um ein Modul genau zu identifizieren. Die Ableitung neuer Klassen für einen Hersteller, teils ohne weitere Properties zu beinhalten, ist gängige Praxis in WBEM/CIM. Auf diese Weise werden Klassen herstellerspezifischen Providern zugeordnet.

Die Modellierung des Systems „TUDIC_Device" als Komposition von logischen Geräten ließe sich noch verfeinern. Für einen Aspekt von Ethernet Ports ist dies im Anwendungsfall Topologie (Abschnitt 6.3) umgesetzt. Weitere Detaillierungen wurden jedoch nicht vorgenommen, da für eine Instanziierung Herstellerwissen notwendig wäre. Ohne die Möglichkeit, zielgerichtet Provider für die Instanziierung entwickeln zu können, ist eine Validierung jedoch ausgeschlossen. Beispiele sind hier die Abbildung von Prozessoren, Pufferbatterien, Spannungsversorgungen, Sensoren etc. CIM sieht für diese Beispiele jedoch bereits Klassen vor, die lediglich automations- bzw. herstellerspezifisch ausgeprägt werden müssen.

6.2 Versions-Management

In diesem Abschnitt werden die einzelnen Modellierungsschritte bzw. Modellentscheidungen, die zum Realisieren des Anwendungsfalls (vgl. Abschnitt 2.1) notwendig sind, genauer betrachtet. Da es sich beim Versions-Management um den ersten Anwendungsfall handelt, wird mit dem Ziel die Komplexität der abgebildeten Informationszusammenhänge zu illustrieren, eine größere Detailtiefe für die Betrachtungen gewählt. In den anschließenden Beispielen wird auf eine so große Detailtiefe verzichtet.

Die in Abbildung 3 dargestellte Prozesskette geht zunächst einmal davon aus, dass alle gegenwärtig erreichbaren Teilnehmer bestimmt werden. Dafür eignet sich die in 6.1 eingeführte Klasse „TUDIC_Device". Eine Enumeration auf dieser Klasse liefert alle Instanzen der erreichbaren

Feldgeräte. Für den gewählten Anwendungskontext sind das alle erreichbaren PROFINET IO Geräte.

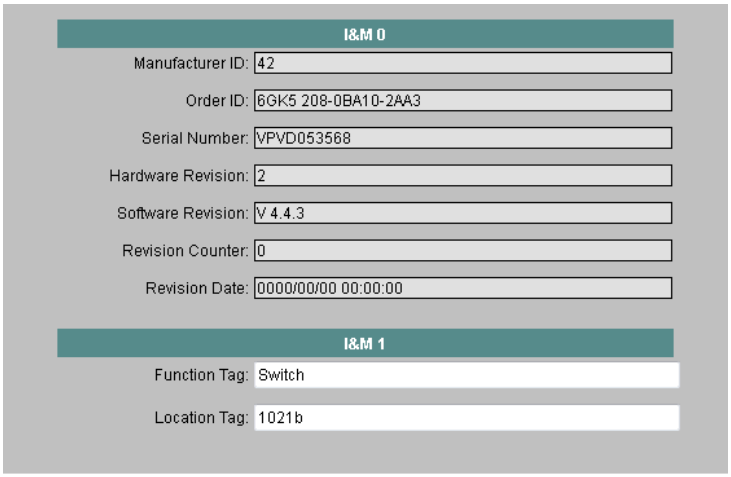

Abbildung 11 I&M Daten im Webfrontend eines Siemens Gerätes

Um den aktuellen Stand der im jeweiligen „TUDIC_Device" vorhandenen Firmwareversion zu ermitteln, fehlen jedoch noch einige Informationen, nicht zuletzt die Firmwareversion selbst. Allein die Informationen in „TUDIC_Device" (vgl. Abbildung 10) reichen dafür offenbar nicht aus.

Für PROFINET IO eignen sich zur Ermittlung dieser Informationen die I&M (genauer die I&M 0) Daten, die jedes PROFINET IO Gerät laut Spezifikation unterstützen muss. Abbildung 11 stellt die in den I&M 0 Daten enthaltenen Informationen anhand eines Beispiels aus der Web-Management-Oberfläche eines Gerätes dar. Im Modell werden I&M-Daten als Spezialisierung von „TUDIC_DeviceIdentification" abgebildet. Die Vererbungshierarchie bis einschließlich „CIM_SettingData" ist in Abbildung 12 dargestellt. „TUDIC_DeviceIdentification" ist dabei die abstrakte Oberklasse für Identifikationsmechanismen. Neben PROFINET IO I&M könnten hier z.B. auch Informationen des CIP Identity Object [166] abgebildet werden, wie es in Kommunikationstechnologien der ODVA verwendet wird.

I&M ist selbst Teil der PROFINET IO Spezifikation und nicht herstellerspezifisch. Daher sind an dieser Stelle Spezialisierungen bzgl. des Herstellers nicht zwingend notwendig.

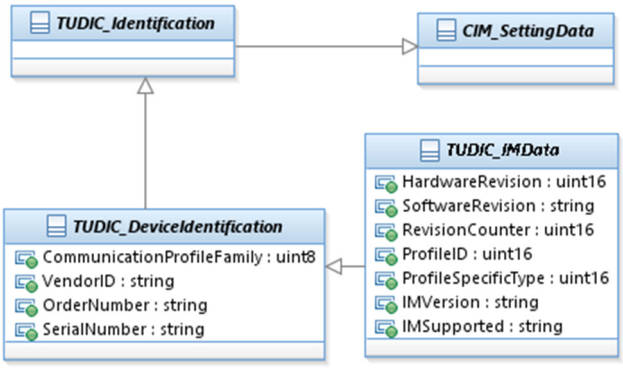

Abbildung 12 Darstellung Identifikationsinformationen

Instanziiert würden hier „TUDIC_IMData" inklusive der geerbten Properties (vor allem VendorID, OrderNumber und SerialNumber) durch einen Provider. Um diese Aufgabe erfüllen zu können, werden vom Provider noch weitere Informationen benötigt, etwa die MAC-Adresse einer spezifischen Geräteinstanz. Diese Informationen können als Instanzen anderer Klassen des Modells, etwa aus „TUDIC_LAN-Endpoint", welche über die Assoziation „TUDIC_HostedProtocolEnd-point" mit jedem „TUDIC_Device" verbunden ist, bezogen werden. Einer „TUDIC_Device"-Instanz können durchaus mehrere Instanzen von „TUDIC_DeviceIdentification" zugeordnet sein. Wie in 6.1 beschrieben, sind I&M-Daten eigentlich Modulen zugeordnet und ein „TUDIC_Device" besteht wiederum aus einer Anzahl an logischen Geräten.

Die Zuordnung von Instanzen der Klasse „TUDIC_DeviceIdentification" zu Instanzen von „TUDIC_Device" erfolgt über die Assoziation „TUDIC_-DeviceIdentification", zu Instanzen von „TUDIC_ATModule" dement-sprechend über „TUDIC_ModuleIdentification". Beide Assoziationen erben von „CIM_ElementSettingData".

Es sind im Anwendungsfall somit alle erreichbaren PROFINET IO Geräte sowie ihre I&M Daten bekannt. Aus den I&M Daten lassen sich nun direkt Informationen zur installierten Softwareversion, sowie die genaue Bezeichnung der Geräteklasse und des Herstellers entnehmen. Mit diesen Informationen kann nun die Klasse „TUDIC_PNSoftwareIdentity" (vgl. Abbildung 13) instanziiert werden. „TUDIC_SoftwareIdentity" dient wiederum als Abstraktionsebene für verschiedene Technologien. Die

Zuordnung von „TUDIC_SoftwareIdentity“ zu „TUDIC_Device“ erfolgt wie dargestellt über die Assoziation „TUDIC_InstalledIndustrialSoftwareIdentity“, welche, wie der Name schon andeutet, zum Ausdruck bringt, dass diese Software gegenwärtig auf dem Device installiert ist. Die generelle Eignung von Software für ein Element (in diesem Fall Device oder Module) wird durch „TUDIC_ElementSoftwareIdentity“ ausgedrückt. Auch hier beziehen die Provider zum Instanziieren der SoftwareIdentity Klassen ihre weiteren Informationen aus „TUDIC_-DeviceIdentfication“ Instanzen.

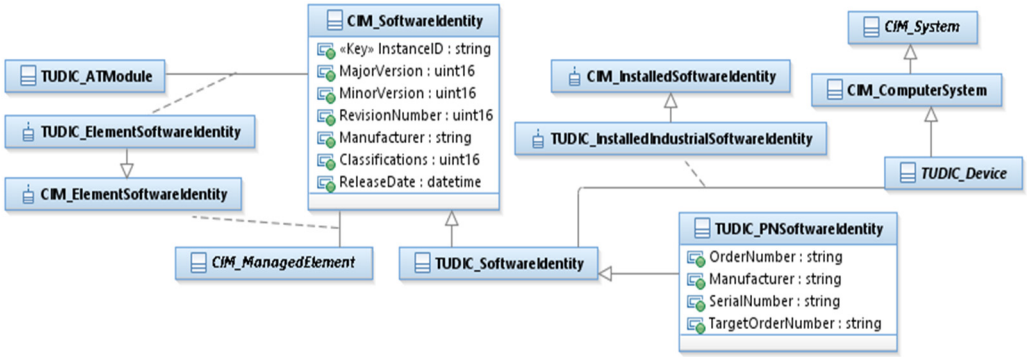

Abbildung 13 Software Identity mit Beziehungen und Abhängigkeiten

Offen ist, (gemäß Abbildung 3) welche Softwareversionen der jeweilige Hersteller für diese Geräteklasse noch vorsieht und anbietet. Auch hier dienen die Informationen aus den I&M-Daten als Einstieg. In [167] wird die Auflösung der nummerischen Herstelleridentifikation (vgl. Abbildung 11 und Abbildung 12) in präzise Bezeichnungen sowie weiterführende Dienste, wie z.B. eine Webseite mit Softwareversionsständen, beschrieben. Angenommen, Hersteller würden diesen Ansatz bereits flächendeckend einsetzen, so könnten auf diese Weise weitere Instanzen von „TUDIC_SoftwareIdentity“ erzeugt werden. Für diese Instanzen würde dann jedoch ausschließlich die Assoziation „TUDIC_-ElementSoftwareIdentity“ instanziiert.

Damit sind aus Sicht des Modells alle Informationen zum Versions-Management von Firmwareständen abgebildet. Ob durch die jeweiligen Provider auch direkt die Abbilder der Softwarestände bezogen werden, bleibt hier offen. Ein möglicher Ansatz findet sich hier in den „Verwaltungsschalen“ des Referenzarchitekturmodells Industrie 4.0

(RAMI) [152] . Wie nun die Auswahl des Soll-Standes der Firmware geschieht, liegt im Wesentlichen beim Client, ebenso der Detailvergleich der Unterschiede zwischen Versionen (Changelog, Patchnotes). Im einfachsten – wenn auch nicht als Best-Practise zu bezeichnenden (vgl. Abschnitt 2.1) – Fall würde der Client die höchste Versionsnummer als neuen Soll-Stand auswählen. Die Installation der so ermittelten Wunschversion könnte über eine Spezialisierung von „CIM_Software-InstallationService" bzw. den darauf definierten Methoden erfolgen. Auf eine Umsetzung musste verzichtet werden, da in der Praxis kein generischer Weg besteht dies auch durchzuführen. Dies betrifft vor allem die Geräteinstanzen, die zum Zeitpunkt der Entwicklungen in der Laborumgebung zur Verfügung standen.

Alle Betrachtungen, die bis zu diesem Punkt für den Anwendungsfall Version-Management angestellt wurden, vernachlässigen, dass ein Versions-Management natürlich auch für „offline"-Geräte (z.B. Ersatzteile) oder nur einzelne Module sinnvoll sein kann. Dies wird allerdings vom Modell implizit bereits abgedeckt und stellt sich letztlich nur als Problem der Informationsbeschaffung, also der Provider-Implementierung und der generellen Informationsverfügbarkeit dar. Sind Geräte offline, kann das Beziehen von Informationen zur aktuellen Firmwareversion in einer Geräteinstanz offenbar nicht über den Online-Zugriff auf die Identifikationsinformationen erfolgen. Es müssten in diesem Fall andere Wege, wie die Einbindung einer Inventar-Datenbank, beschritten werden. Auf die beschriebenen Modellteile hätte dies jedoch keinen direkten Einfluss.

Das Management von Softwareversionen ist natürlich nicht auf Firmware beschränkt. Es ist nicht ausgeschlossen, auch verschiedene PLC-Programme über ein übergeordnetes Management zu verwalten. Dies ist jedoch nicht Bestandteil dieser Arbeit (vgl. 5.3.2(iii)).

6.3 Topologie-Erkennung

Die Komplexität dieses Anwendungsfalls (vgl. Abschnitt 2.2) ist eher niedrig. Im Verlauf der Arbeiten mit den entwickelten Modellteilen und Prototypen, z.B. im Projekt SMartA [58], stellte sich jedoch heraus, dass für viele weitere Anwendungsfälle die Erkennung der Topologie essentiell ist. Die Bewertung der Verbindungsqualität zwischen zwei Kommunikationspartnern ist z.B. nur sinnvoll möglich, wenn auch bekannt ist, auf welchem Wege diese beiden Partner miteinander in Verbindung stehen. Für die Ableitung von Netzwerkdiagnosen, wie in [102] beschrieben, ist eine korrekte Topologie ebenfalls notwendig um eine Lokalisation durchführen zu können.

In diesem Anwendungsfall soll die Topologie-Erkennung mit Netzwerkunterstützung erfolgen (vgl. Abschnitt 2.2). Für Profinet-basierte Netzwerke bietet sich hier die Nutzung der Informationen aus LLDP [168] an, da Geräte aller Konformitätsklassen diese Unterstützung bieten müssen. LLDP-Informationen werden zwischen Geräten ausgetauscht und können dann entsprechend als Einträge in der jeweiligen Geräte-MIB per SNMP abgefragt werden.

LLDP stellt eine ganze Reihe von Informationen zur Verfügung, die jeweils helfen die Geräte selbst auf Layer 2 zu identifizieren. Zusätzlich bietet die LLDP-MIB Informationen die Kommunikationspartner, welche über einen bestimmten Port mit der lokalen Geräteinstanz verbunden sind, zu ermitteln. Auf Basis von Informationen zur lokalen und entfernten Geräteidentifikation kann somit die Topologie des Netzes nachgebildet werden.

In Abbildung 14 werden einige der Informationen, die per LLDP zu Verfügung stehen, dargestellt. Das organisatorische Vorgehen zur Ermittlung der Topologie erfolgt dabei analog zur Abbildung 4.

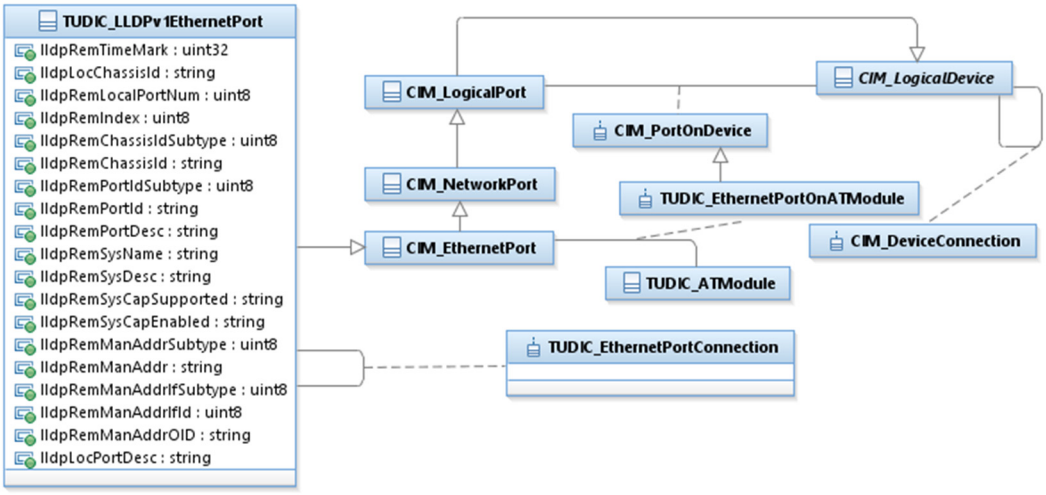

Abbildung 14 Strukturen für die Bestimmung der Topologie

Die Ermittlung der erreichbaren Teilnehmer erfolgt grundsätzlich wie in 6.1 beschrieben, über die Instanzen von „TUDIC_Device". Geht man davon aus, dass sich im relevanten Netzwerksegment nicht nur automatisierungstechnische Geräte mit Ethernet-Anbindung befinden, so ist es sinnvoll, die Ermittlung der erreichbaren Teilnehmer nicht über „TUDIC_Device", sondern bereits eine Abstraktionsebene darüber durchzuführen. Auf diese Weise wird eine vollständige Liste aller Systeme, beinhaltend auch Desktop PC, Switches usw., gewonnen. Auf das generelle Vorgehen in der Folge hat dies jedoch, unter der Annahme, dass für alle Instanzen von „CIM_ComputerSystem" entsprechende „CIM_ProtocolEndpoints" instanziiert werden können, keinen Einfluss.

Mit der nun vollständigen Liste aller Netzwerkteilnehmer können die LLDP-Informationen pro Ethernet-Anschluss (Abbildung 14) abgebildet werden. „TUDIC_LLDPv1EthernetPort" bzw. der zugehörige Provider erzeugt für jeden Netzwerkanschluss, sofern er eigenständig ist, eine Instanz. Diesen Instanzen sind alle Spezialisierungen der Klasse „CIM_LogicalDevice" und über die Assoziation „TUDIC_EthernetPortOn-ATModule" einem Modul und somit indirekt auch einem „TUDIC_Device" zugeordnet.

Nachdem die LLDP-Informationen in CIM abgebildet sind, muss noch die Zuordnung der Verbindung zwischen einzelnen Komponenten des

Netzwerkes erfolgen. Praktisch geschieht dies durch die Instanziierung von „TUDIC_EthernetPortConnection", wodurch jeweils zwei Logical-Devices, in diesem Fall benachbarte Ethernet-Ports, in Relation gesetzt werden. Somit ist die einfache Netzwerktopologie abgebildet.

Um die Informationen – vor allem LLDP-Informationen –, die zum Instanziieren der genannten Klassen notwendig sind, zu beschaffen, bleibt, sofern dies von einer zentralen Stelle aus geschieht, nur SNMP. Welche OID (SNMP) die Informationen beinhaltet, die der entsprechenden CIM-Property zuzuordnen sind, kann auf Modellebene über sogenannte MappingStrings [92] erfolgen. Entsprechend würde der Qualifier MappingStrings von „TUDIC_LLDPv1EthernetPort.lldpRem-ChassisID" MIB.IEEE|LLDP-MIB.lldpRemChassisId als Wert enthalten. Auf diese Weise kann in CIM direkt in andere Informationsmodelle referenziert werden. Es ist ohne weiteres möglich, aufbauend auf den Informationen der MappingStrings, generische Provider zu erzeugen, die ohne weiteren Aufwand die Beschaffung der Informationen über SNMP umsetzen.

Es ist für diesen Anwendungsfall notwendig, umfassend auf die Nutzung von SNMP zurückzugreifen. Vollkommen unkritisch ist dies nicht, da aus Sicht der eigentlichen Management-Aufgabe auf den ersten Blick eher Komplexität (SNMP+WBEM/CIM) hinzugefügt als abgebaut wird. Für die einfache Beschaffung der LLDP-Informationen ist dies noch bedingt richtig, auch wenn z.B. das initiale Auffinden aller Geräte per SNMP nur durch scannen des Netzes möglich ist. Für die Herstellung der Beziehungen zwischen den Geräten – also der Topologie – und somit der eigentlichen Management-Aufgabe sinkt tatsächlich die Komplexität durch den Einsatz von WEBM/CIM. Für die Aufgabe der Zuordnung benachbarter Ethernet-Ports wird im NMS keinerlei Logik mehr benötigt, wie es bei der reinen Nutzung von SNMP der Fall wäre. Dies eröffnet Herstellern neue Möglichkeiten und bietet Endanwendern die Option auf generische Clients zurückzugreifen. Die Nutzung von SNMP, vor allem der damit einhergehenden Netzwerkkommunikation, kann unter der Voraussetzung, dass Provider bzw. WEBM/CIM-Server direkt in den

automatisierungstechnischen Geräten betrieben werden (vgl. Abschnitt 5.2), wesentlich reduziert werden.

6.4 Alarm-Handling

Das Handling von Alarmen in Anlagen mittels WEBM/CIM ist der komplexeste hier diskutierte Anwendungsfall. Alle in den vorangegangenen Abschnitten beschriebenen Anwendungsfälle stehen mit dem Alarm Handling in Beziehung. Entweder müssen sie wie im Fall der Geräteabbildung (vgl. Abschnitt 6.1) direkt herangezogen werden oder auf ihrer Basis können in der Folge weitere Informationen abgeleitet werden. Im Zusammenhang mit dem Versions-Management (vgl. Abschnitt 6.2) kann dies beispielsweise das Erkennen eines vermehrten Auftretens von Alarmen nach der Änderung einer Softwareversion sein. Die Kenntnis der Topologie (vgl. Abschnitt 6.3) wiederum ermöglicht die genaue Lokalisation von Alarm und das Erkennen von Änderungen an der Topologie.

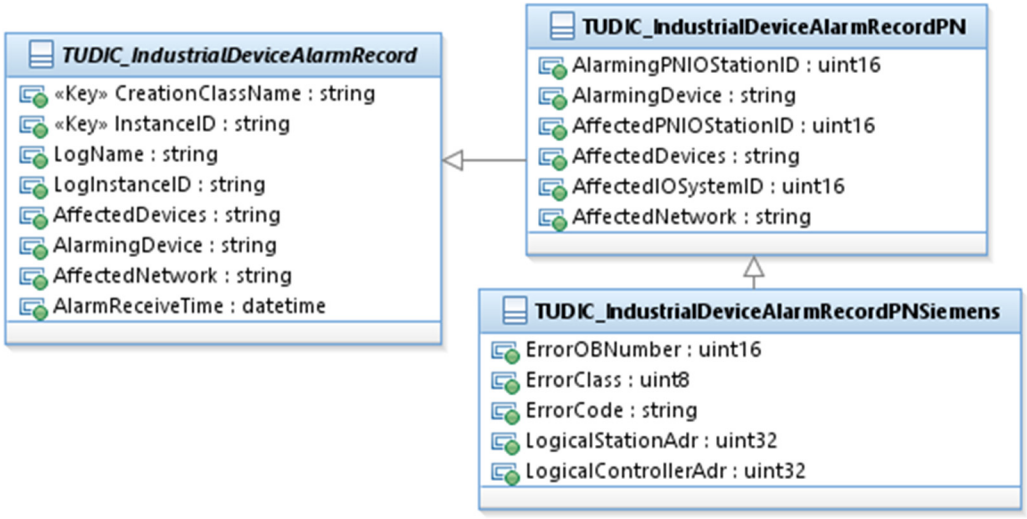

Abbildung 15 Abbildung von Alarmen in CIM

Dargestellt in Abbildung 15 ist eine Möglichkeit, Alarme industrieller Geräte in CIM abzubilden. Auffällig ist hier wieder die Spezialisierung nach Feldbus und Hersteller. „TUDIC_IndustrialDeviceAlarmRecordPN-Siemens.ErrorOBNumber“ hat zum Beispiel nur im Kontext von

Umgebungen des Herstellers Siemens eine Bedeutung, da Organisationsbausteine (OB) in aller Regel herstellerspezifisch sind. In der Darstellung ebenfalls ersichtlich ist die Trennung nach Geräten, die durch den Alarm betroffen sind – „AffectedDevices" – und dem Alarm registrierenden Gerät z.B. einer SPS in Form des „AlarmingDevice". Das Instanziieren der dargestellten Klassen kommt dem Registrieren eines einzelnen Alarmes (vgl. Abbildung 5) im System gleich.

In der Praxis besteht die größte Herausforderung bei der Umsetzung des Alarm-Handlings in externen Management-Systemen darin, dass Feldbus-Organisationen hierfür keinen einheitlichen Weg definieren. Häufig werden Alarme bzw. Meldungen innerhalb der Grenzen des jeweiligen Systems mit Hilfe von herstellerspezifischen Werkzeugen behandelt. Die hierarchische Weitergabe von Alarmen an übergeordnete Systeme ist jedoch nicht grundsätzlich ausgeschlossen. Das in Abschnitt 4.4 beschriebene OPC UA (speziell OPC UA: Alarms and Conditions [130]) ist hierfür ein Bespiel.

Wie die eigentlichen Alarminformationen im Rahmen dieser Arbeit aus dem realen System in das Informationsmodell transferiert werden (Pull vs. Push-Mechanismen im Provider), wird in [102] beschrieben. Für die Instanziierung des Modells ist es grundsätzlich unerheblich, ob die Alarminformationen aus den Feldgeräten gelesen werden oder ob Alarme durch die jeweilige SPS an das Management-System veröffentlicht werden.

Bis zu diesem Punkt werden lediglich beliebige Alarme durch das Management-System registriert. Eine Zuordnung oder Gruppierung lässt sich durch die in Abbildung 15 dargestellten Klassen jedoch noch nicht erkennen. Die in Abbildung 16 dargestellte Klasse „TUDIC_AlarmLog" trägt nun dafür Sorge, dass alle Alarme, die in einer SPS anliegen, in einem Logbuch zusammengefasst werden. Durch „TUDIC_AlarmLog" ist neben der Gruppierung auch das Löschen aller Logeinträge umgesetzt. Die Zuordnung zwischen Logbuch und Alarm erfolgt durch „TUDIC_-LogManagesRecord".

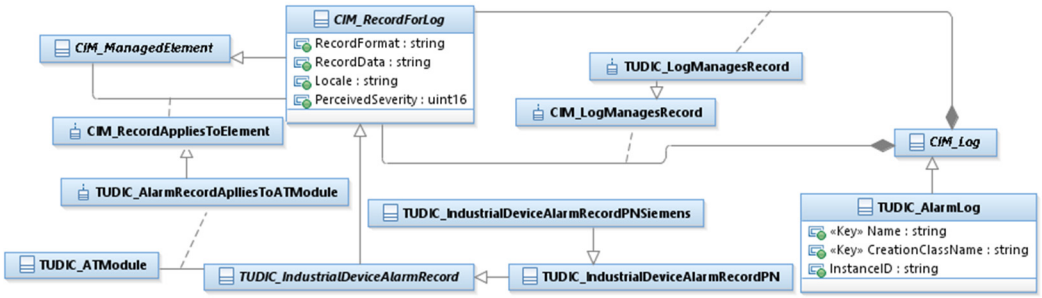

Abbildung 16 Einordnung von Alarmen in ihren Kontext

Wie in den anderen Anwendungsfällen stehen auch hier die „TUDIC_"-Klassen nicht eigenständig, sondern sind Spezialisierungen vorhandener CIM-Klassen gemäß der Bedürfnisse der Automation.

Um die einzelnen Alarme einem spezifischen Automatisierungsgerät zuordnen zu können, müssen diese Geräte zunächst wieder abgebildet werden. Dies ist durch 6.1 bereits vollständig abgedeckt. Alle relevanten Geräte sind zu diesem Zeitpunkt durch eine Instanz von „TUDIC_ATModule" repräsentiert. Die Zuordnung eines Alarmes zu den direkt betroffenen Geräten kann durch Instanzen von „TUDIC_Alarm-RecordAppliesToATModule" erfolgen. Da als Festlegung dieser Arbeit Alarme immer im Kontext eines Gerätes auftreten, spezialisiert „TUDIC_AlarmRecordAppliesToATModule", von der Zuordnung von Alarmen auf allgemeine „CIM_ManagedElement" durch „CIM_Record-AppliesToElement", auf die Zuordnung von Alarmen zu Logical Devices bzw. genauer „TUDIC_ATModule". Durch die Instanziierung der Assoziationen ist die in Abbildung 5 dargestellte Prozesskette abgeschlossen.

Wie in der Folge weiter mit den abgebildeten Alarmen verfahren wird, bleibt in dieser Arbeit weitestgehend offen. Im Zusammenhang mit anderen diskutierten Anwendungsfällen lassen sich unter Nutzung der CIM Query Language (CQL) [169] weitere Informationen auf Basis registrierter Alarme ableiten. Beispiele hierfür wurden bereits angeführt. Des Weiteren ist es in der Automation üblich, Alarme nicht einfach nur zu registrieren, sondern, abgesehen von etwaigen Problemlösungen, zu quittieren. Hierfür auf ein übergeordnetes System-Management zu vertrauen, kann besonders für direkt prozessrelevante Alarme kritisch

sein, weshalb bereits in Abschnitt 2.3 davon abgesehen wurde, die Alarmbehandlung – im Sinne eines Quittierens von Prozessalarmen – mit dem vorgestellten Anwendungsfall abzudecken. In der Zukunft ist es jedoch nicht ausgeschlossen Alarme, die primär die Infrastruktur und nicht vordergründig den Betrieb des Prozesses betreffen, teilweise auch innerhalb des vorgestellten Ansatzes zu behandeln. So könnte z.B. das Behandeln von Alarmen, die mit dem Ausfall eines Switches und dem damit verbundenen automatischen Wechsel auf alternative Topologien durch die Nutzung redundanter Kommunikationspfade in Zusammenhang stehen, in Zukunft innerhalb des System-Managements quittiert werden. Technisch wäre dies dann äquivalent zu heute in der Automation eingesetzten Quittierungsmechanismen (z.B. für Switches die als PROFINET IO Gerät betrieben werden können), allerdings mit dem Vorteil eines generischen System-Managements.

7 Evaluierung

Eine quantitative Bewertung der Ergebnisse bzw. Modellerweiterungen als Ganzes sowie der praxisnahen Leistungsfähigkeit (in Form der Instanziierungen) ist kaum machbar. Zur vergleichenden Bewertung untereinander weisen die vorgestellten Anwendungsfälle ein zu hohes Maß an Heterogenität auf. Sie bauen zwar teilweise aufeinander auf, sind aber von ihren Zielen und Ansätzen zur technischen Umsetzung vollkommen verschieden. Eine Metrik zur Bewertung müsste demzufolge unabhängig für jeden Anwendungsfall erstellt werden.

Eine quantitative Bewertung der Ergebnisse im Ganzen gegenüber anderen etablierten Ansätzen (z.B. OPC UA oder SNMP) scheitert daran, dass es mit keinem anderen Ansatz möglich ist die Anwendungsfälle in Gänze vergleichbar umzusetzen. In der Regel scheitert ein Vergleich an technischen oder organisatorischen Details.

Beispiele:

> Das Handhaben von automatisierungstechnischen Alarmen mit den Mitteln von SNMP ist im Rahmen der durch das Protokoll und Informationsmodell eingeräumten Möglichkeiten prinzipiell möglich. Es existiert jedoch keine bekannte praktische Umsetzung, die als Vergleichsbasis dienen könnte. Hinzu kommt, dass Teilaspekte des Anwendungsfalls, wie die Zuordnung von Alarmen zu Geräten, modellbedingt gar nicht effizient umzusetzen sind.

Das Management von Softwareversionen mittels SNMP ist mit Hinblick auf den hier beschriebenen Anwendungsfall ebenfalls schwierig. Viele der herangezogenen Informationen sind in den MIBs der Geräte in aller Regel nicht vorhanden. Andere Aspekte des Anwendungsfalls, z.B. die Abbildung von verfügbaren Softwareversionen im Informationshaushalt, sind in SNMP schon technisch nahezu ausgeschlossen, zumindest jedoch außerhalb aller bekannten Spezifikationen. Für diesen Anwendungsfall wurde auf technischer Ebene weitgehend auf PROFINET IO-eigene Kommunikationsmöglichkeiten vertraut.

OPC UA, welches in Bezug auf das System-Management konzeptionell vergleichbar leistungsstark wie WBEM/CIM sein kann (vgl. Abschnitt 4.4), scheidet als Vergleichspartner für eine quantitative Bewertung aller das Modell betreffenden Erweiterungen ebenfalls aus. Wesentlich ist hier, dass OPC UA, bzw. dessen Informationsmodell, zum Zeitpunkt des Entstehens dieser Arbeit keine relevante Ausprägung in Bezug auf das System-Management aufweist (vgl. Abschnitt 4.4 und 4.8).

Da eine durchgängige quantitative Bewertung der Ergebnisse nicht möglich ist, muss qualitativ bewertet werden. Nur an einem nachvollziehbaren Beispiel, das einen Teilaspekt eines Anwendungsfalls darstellt, wird eine quantitative Bewertung durchgeführt

Was die Fähigkeit der vorgestellten Modellerweiterungen zur Lösung der jeweiligen Anwendungsfälle angeht kann nur gesagt werden, dass es sich jeweils um *eine* Lösung handelt. Es existieren andere. Ob diese mehr oder weniger geeignet sind müsste im Einzelnen bewertet werden. Die vorgestellten Modelle sind jedoch über einen längeren Zeitraum durch die Arbeit an Projekten und Veröffentlichungen entwickelt worden, so dass davon ausgegangen werden kann, dass sie eine gewisse Reife besitzen.

Die Kernfrage, die sich während der Erstellung einer Erweiterung für CIM immer stellt ist: Inwiefern ordnet sich die Erweiterung in die „CIM-

Lesart" ein. Neben der Identifikation der am besten geeigneten Superklasse für die jeweilige Erweiterung stellten sich bei CIM generelle Fragen die mit der objektorientierten Modellierung in Zusammenhang stehen. „Als lokale Property oder andere/eigene Klasse", „wohin gehören Properties" und „neue oder spezialisierte Assoziation" sind nur einige Beispiele für diese Fragen, die während der Modellierung immer wieder auftreten. Beantwortet werden können diese nur nach einem gewissen Lernprozess und selbst dann ist nicht sicher, ob die Antwort über einen Zeitverlauf dieselbe bleibt. Beispielsweise waren in ersten Versionen von „TUDIC_DevicePN" (vgl. Abschnitt 6.1) die IP-Adresse sowie die MAC schlicht Properties der Klasse, erst im späteren Verlauf erwies sich es sich als notwendig und auch sinnvoll diese beiden Informationen jeweils als „ProtocolEndpoint" darzustellen. Was an dieser Stelle klar wird, das Darstellen von neuen Informationen ist nicht trivial und um effizient und gut wiederverwendbar in CIM zu modellieren ist nicht nur die Kenntnis der Domäne äußerst wichtig, sondern auch ein fundiertes Wissen über die vorhandenen CIM-Core- und –Common-Schemata notwendig. Hier liegt in der Anwendung von CIM in der automatisierungstechnischen Praxis ganz klar eine Herausforderung. Eine Möglichkeit dieser gezielt zu begegnen wäre die Spezifikation von automationsspezifischen Profilen. Für die IT existieren von Seiten der DMTF bereits Profile für viele übliche Anwendungsfälle, diese könnten der Einstiegspunkt für vergleichbare automationsspezifische Profile sein.

WBEM/CIM ist mit allen anderen Management-Ansätzen gemein, dass sie vorrangig von ihrer Verbreitung und weniger von ihrer prinzipiellen Leistungsfähigkeit abhängig sind, wenn es um Belange des Einsatzes im Feld geht. SNMP als „Zwischenlösung" [170] hat dies in den vergangenen Jahren gezeigt. Für den Bereich der Enterprise-IT besitzt WBEM/CIM eine hinreichende Verbreitung. In der Automatisierungstechnik ist diese Verbreitung bislang eher unbewusst und stellt sich in Form von IT-Hardware an den Übergängen zu Unternehmensbereichen (z.B. ERP) oder im Feld eingesetzter IT-Software (z.B. Microsoft Windows) dar. Ob es nun WBEM/CIM oder doch eher OPC UA ist, welches zukünftig für das System-Management eingesetzt wird, wird die Zeit zeigen müssen.

Gegenwärtig eignet sich WBEM/CIM für die reinen Management-Aufgaben jedoch – nach den Untersuchungen dieser Arbeit – besser. Es bleibt abzuwarten, ob der Integrationswille in der Automation stark genug ist, um diesen Weg zu gehen und eine IT-Technologie für die Automation zu adaptieren oder ob man weiterhin nahezu ausschließlich auf automationsspezifische Ansätze vertraut. Viele der Überlegungen und Konzepte in Industrie 4.0 würden jedoch für eine Integration sprechen.

Die qualitative Bewertung der Anwendungsfälle kann jedoch als Indikator für die generelle Leistungsfähigkeit der Modellerweiterungen und somit für die Potentiale für WBEM/CIM in der industriellen Automation herangezogen werden. Der Vergleich auf Ebene des Netzwerkes wurde bereits durchgeführt. Um die Leistungsfähigkeit aus Sicht des Anwenders zu bewerten bietet es sich an, die Anzahl an notwendigen Anfragen, die ein Nutzer zur Bearbeitung der jeweiligen Management-Aufgabe stellen muss, zur vergleichenden Bewertung heranzuziehen.

Das Versions-Management ist mit den Mitteln von SNMP gegenwärtig nahezu ausgeschlossen. Es existiert, abseits von Enterprise-MIBs, keine einheitliche Beschreibung der benötigten Informationen (z.B. Hersteller, Seriennummer, Software Version). Selbst für den Fall, dass sie existieren stellt sich für den Anwender die Frage, wie er an diese Informationen gelangen soll. So waren z.B. die Enterprise-MIBs der Firma Siemens nach letztem Stand gar nicht ohne weiteres für alle Aspekte verfügbar. Hierdurch wird auch noch einmal verdeutlicht, dass es durchaus wünschenswert ist, das Modell selbst aus dem im Betrieb befindlichen Management-System beziehen zu können.

Einmal angenommen, die entsprechenden Informationen würden in geeignetem Umfang zur Verfügung stehen, so bedürfte es pro Gerät einer „SNMP GETBULK“ Anfrage um diese Informationen aus den Feldgeräten zu beschaffen. Das im Vorfeld die vorhandenen Geräte und damit auch ihre Adressen im Netzwerk überhaupt erst einmal bekannt sein müssen ist ebenfalls zu beachten. Als Ergebnis steht eine Liste mit

Geräten und ihren zugehörigen Versionsinformationen zur Verfügung. Um die inhaltlich vergleichbare Liste unter Verwendung des in Abschnitt 6.2 erläuterten Modells zu erhalten genügt die Enumeration aller Instanzen der Klasse „TUDIC_InstalledIndustrialSoftwareIdentity". Dabei ist hier unerheblich, wie viele Geräte sich tatsächlich im System befinden, es bleibt immer eine einzelne Anfrage (Abbildung 17).

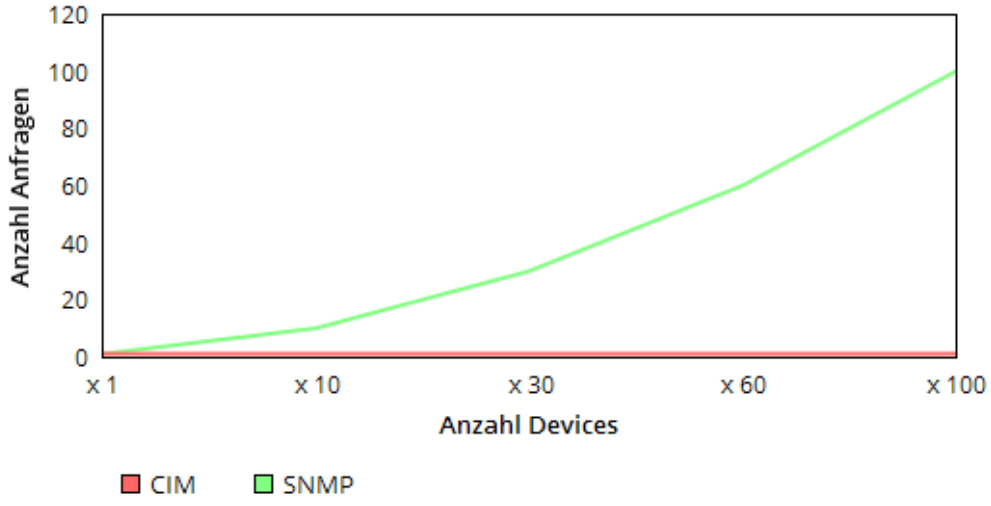

Abbildung 17 Anzahl der Anfragen aus Nutzersicht

Der technische Aufwand ist natürlich nicht reduziert. Der wesentliche Teil dieses Aufwandes wird aber vom Nutzer auf das Modell übertragen. Schon für diesen Teilaspekt des Anwendungsfalls Versions-Management zeigt sich das Potential von CIM für den Endnutzer. Alle weiteren Schritte (vgl. Abschnitt 6.2) sind rein mit SNMP gar nicht mehr möglich, es muss zwingend auf Funktionen eines NMS vertraut werden. Durch den Einsatz von WBEM/CIM ist prinzipiell der gesamte Anwendungsfall innerhalb eines durchgängigen und auf abstrakter Ebene herstellerunabhängigen Ansatzes möglich. Durch die Definition spezieller „Firmware-Update-Views" in CIM kann der gesamte Prozess für den Nutzer sogar noch weiter zusammengefasst werden. Die wesentlichen Effekte durch die Nutzung von WBEM/CIM gegenüber SNMP sind für das Versions-Management ganz klar:

- die Behandlung einer Management-Aufgabe die sich über mehrere Technologieebene bzw. Protokolle erstreckt
- die Konsolidierung verschiedener Informations- und Datenquellen
- die Unabhängigkeit, was die zu wählende Client-Software zur Lösung der gegebenen Management-Aufgabe angeht

Die Zusammenhänge zwischen WBEM/CIM und SNMP im Anwendungsfall Topologie-Erkennung zeichnen sich klarer ab, da hier die technische Umsetzung der WBEM/CIM Provider in wesentlichen Aspekten selbst auf SNMP zurückgreift. In beiden Fällen wird die LLDP-MIB aus den jeweiligen Geräten gelesen. Für den direkten Weg über SNMP gilt analog zum Versions-Management (siehe auch Abbildung 17) für jedes Gerät muss durch den Nutzer eine Anfrage gestellt werden. Um vergleichbare Ergebnisse über WBEM/CIM zu erreichen genügt es die Instanzen von „TUDIC_LLDPv1EthernetPort" zu enumerieren. Der wesentliche Anteil dieses Anwendungsfalls, das Zuordnen benachbarter Ethernet-Ports, muss für den direkten Weg über SNMP wiederum im externen NMS geschehen und lässt sich nicht auf Basis des Modells abbilden. Anders bei der Verwendung von WBEM/CIM hier genügt wiederum das Enumerieren von „TUDIC_EthernetPortConnection" um alle Nachbarschaftsbeziehungen und somit die Topologie zu erhalten.

Die wesentlichen Effekte die, durch den Einsatz von WBEM/CIM im Anwendungsfall Topologie-Erkennung erzielt werden können, sind also:

- die Abbildung der Topologie bereits auf Modellebene, durch das Herstellen der Port-Port bzw. Gerät-Gerät Beziehungen.
- die daraus resultierende Unabhängigkeit des Managements von spezifischen Clients
- eine Reduktion der Anfrage-Anzahl sowie der Wegfall des Prozesses zum Auffinden von Geräten aus Sicht des Endanwenders bezogen auf SNMP

Kann für die SNMP-Anfragen durch den Nutzer nicht auf SNMP-BULK zurückgegriffen werden so fällt die Reduktion der Anfragen aus Nutzersicht noch dramatischer aus (vgl. Abbildung 18). Praktisch kann

dies in der Automation auch heute noch der Fall sein da, wenigstens bis zum Beginn dieser Arbeit, noch Geräte vertrieben wurden die lediglich SNMPv1 unterstützten.

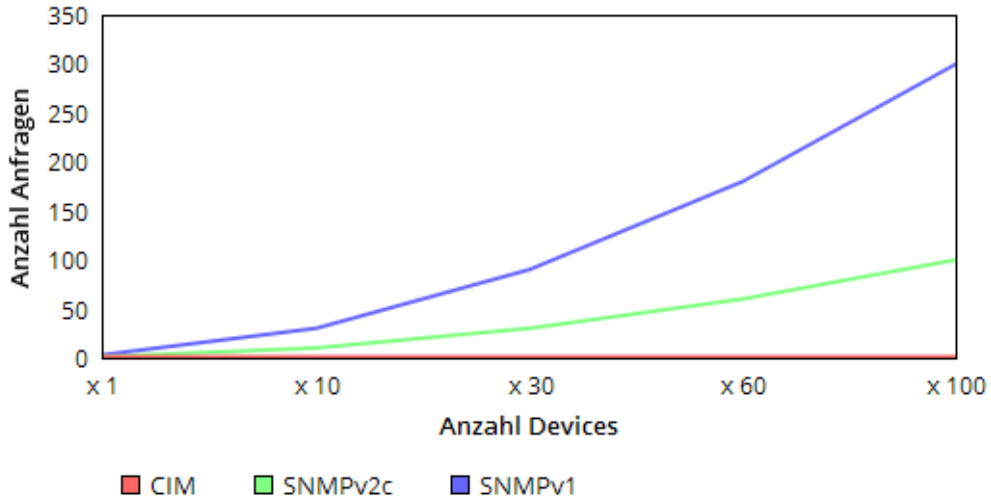

Abbildung 18 Anfrageanzahl ohne SNMP-BULK-Requests

In der Abbildung wird die Entwicklung der Anfragen in einer PROFINET IO Umgebung ohne die Unterstützung von SNMP-BULK-Requests dargestellt. In diesem Fall werden für das Aufspannen der Topologie unter Zuhilfenahme von LLDP mindestens drei SNMP-GET-Requests pro Gerät benötig. Dabei wird angenommen, dass ein durchschnittliches PROFINET IO Gerät drei externe Ethernet-Ports besitzt.

Für den letzten verbleibenden explizit umgesetzten Anwendungsfall, das Alarm Handling, bietet sich der direkte Vergleich mit SNMP nicht an. Es kommt für diesen Bereich, bis auf Ausnahmen, nicht zum Einsatz. Die generelle Problematik von Push und Pull Verfahren zur Extraktion der Alarme aus den Feldgeräten wurde bereits [102] diskutiert. Vor dem Nutzer bleibt dies jedoch ohnehin verborgen, egal ob die Systemweite Alarmbehandlung nun per WBEM/CIM oder durch automatisierungs-spezifische Werkzeuge erfolgt. Die wesentlichen Mehrwerte durch den Einsatz von WBEM/CIM gegenüber klassischen Automatisierungslö-sungen sind in diesem Anwendungsfall:

- Alarm-Handling mit generischen Clients und somit direkte Integration in ein systemweites Management-Konzept
- daraus resultierend die Integration automationsspezifischer Alarme in ein unternehmensweites Fault-Management
- die Persistierung von Alarmen und ihren Relationen zu Systemkomponenten auf Basis des Modells
- Abstraktion über Alarme verschiedener Hersteller und Feldbusse

Die Fähigkeit zur Abstraktion ist generell eines der ganz wesentlichen Potentiale für die Automation. Aus Sicht des System-Managements haben PROFINET IO-, Ethernet/IP- und IT-Netzwerk-Geräte mehr gemein, als sie bezogen auf die Vielzahl der allgemeinen Management-Aufgaben unterscheidet. Für spezifische Management-Aufgaben und natürlich auch, um der Automation den Zugang zu WBEM/CIM zu ermöglichen sind Erweiterungen an CIM notwendig, wie sie in dieser Arbeit durchgeführt wurden.

Wie bereits dargestellt sind die vorgenommenen Modellerweiterungen sicherlich nur eine Möglichkeit die jeweiligen Aspekte in CIM abzubilden. Es gibt hier selten nur einen einzelnen richtigen Ansatz. In der akademischen Praxis konnten sich die konkreten, in dieser Arbeit entwickelten Modelle, jedoch bereits bewähren [100] [101] [102] [58] und haben sich als hinreichend leistungsfähig herausgestellt. Sofern Kenntnisse der bestehenden CIM-Schemata sowie der neu zu modellierenden Aspekte der Automation vorhanden sind, ist das Einbringen von neuen Aspekten in das Modell jederzeit möglich. Allerding ist hier auch die derzeit größte Limitierung zu finden.

Die Komplexitäten sowohl von CIM wie auch der Automatisierungsbranche sind so hoch, das ein einfacher Einstieg in der Breite eine Herausforderung darstellen wird. Solange dieser Einstieg in der Breite jedoch nicht geschehen ist, wird die Automation nicht das volle Potential des Ansatzes ausschöpfen können. Dies gilt natürlich auch und insbesondere für die im Rahmen dieser Arbeit entwickelten Modelle. Das Abbilden einer ganzen Domäne mit all ihren technischen und organisatorischen Eigenheiten ist nicht nur generell eine große Herausforderung,

speziell auf Ebene der allgemeingültigen, abstrakten Eigenschaften muss so modelliert werden, dass die Interessen aller späteren Anwender widergespiegelt und ihre Anforderungen erfüllt werden. Grundlegenden Änderungen an allgemeingültigen Komponenten des Modells sind im Nachhinein nur schwer möglich, da die Gefahr besteht die Kompatibilität mit Bestandssystemen nachhaltig zu beeinträchtigen.

Die Modelle in dieser Arbeit haben nicht den Anspruch, auch nur alle wesentlichen Teile der Domäne abzubilden, dies können sie auf Grund der Komplexität auch gar nicht. Außerdem kann man sie nicht als in der Domäne abgestimmt betrachten. Sie sind primär ein Konzept und Umsetzungsvorschlag. Die allgemeine und breite Akzeptanz sind aber gerade elementar, wenn verhindert werden soll, dass auch auf Ebene des System-Managements unter Verwendung von CIM jeder Hersteller und jede Feldbusorganisation Eigenentwicklungen vorantreibt. So würde es unweigerlich wieder zu Inkompatibilitäten, Redundanzen und neuen Integrationsaufwänden kommen.

Während gegenwärtig eine der wesentlichen Herausforderungen für das System-Management der Automation in der Heterogenität liegt, wird zukünftig die Homogenisierung innerhalb der Automation und die Integration mit bestehenden IT-Strukturen die Aufgabe sein. Hier sind zum einen Standardisierungsgremien aber auch Fachgemeinschaften und Fachvereine, in denen sich Firmen und akademische Einrichtungen engagieren, gefragt, diesen Herausforderungen zu begegnen.

8 Zusammenfassung und Ausblick

Die Arbeit bewertet anhand von abgeleiteten Kriterien in welchem Umfang sich existierende Ansätze zum System- und Dienste-Management aus unterschiedlichen Industrien für die Belange der industriellen Automation eignen. Herausgearbeitet werden die relevanten Kriterien auch anhand von Anwendungsfällen, die als relevant und repräsentativ für die Zieldomäne angesehen werden. Auf Basis der umfangreichen Analyse, aus der hervorgeht, dass WBEM/CIM der am besten geeignete Ansatz ist, werden in der Folge diverse Erweiterungen am Informationsmodell CIM durchgeführt. Dies wird anhand der eingangs aufgestellten Anwendungsfälle veranschaulicht. Zusammenfassend kann gesagt werden, dass alle wesentlichen Ziele der Arbeit (Abschnitt 1.2) erreicht wurden.

An einigen Punkten der Arbeit wurde jedoch bewusst auf Detaillierungen verzichtet. Folglich bieten sich einige Ansatzpunkte für weitere wissenschaftliche Arbeiten. So wurde bereits die Notwendigkeit von entsprechenden Standardisierungen angeführt. Konkret sind das zum einen weitere automatisierungsspezifische Modelle und deren Domänenabstimmung, zum anderen die Entwicklung von Profilen und Views für die Automation. Auch die bearbeiteten Anwendungsfälle lassen jeweils noch Raum für zukünftige Erweiterung bzw. Verfeinerungen (vgl. Abschnitt 2.5). Das Versions-Management kann um die Aktivitäten der Hersteller erweitert werden, allem voran das Bereitstellen von Diensten zum strukturierten Zugriff auf den Firmware-Lifecycle für entsprechende

Feldgeräte. Die Topologie-Erkennung ließe sich zum einen um entsprechende Methoden zur Erkennung ohne explizite Netzwerkunterstützung erweitern, zum anderen liegt hier einiges Potential in der Erstellung von CIM-Views für die exakte Zustands-/Fehlerlokalisation auf Basis des Informationsmodells. Das Alarmhandling ist im Modell gegenwärtig noch eher grobgranular abgebildet, auch hier besteht die Möglichkeit zur Verfeinerung. Weiterhin werden Alarme bzw. Ereignisse gegenwärtig nur aufgenommen und persistiert, die weitere Behandlung und Verarbeitung ist ebenfalls offen geblieben.

Generell bieten die Themenkomplexe Internet of Things, Cyber Physical Systems und Industrie 4.0 vor dem Hintergrund eines integrierten bzw. integrierenden System-Managements noch außergewöhnlich viele Arbeitsfelder. Vor diesem Hintergrund wird es zukünftig jedoch zwingend notwendig sein, Methoden zur Verfügung zu haben, um aus vorhandenen Informationen Modelle für das Management ableiten zu können. In der Automation sind dies vor allem Informationen, die in Form von Gerätebeschreibungen oder auch Daten zum Product Lifecycle vorliegen. Vorgehensmodelle zum Ableiten von Informationsmodellen zum System-Management können hier schon eine wesentliche Beschleunigung des Prozesses mit sich bringen. Diese gilt es zu entwickeln. Eine noch stärkere Vereinfachung des Prozesses würde erreicht, wenn eine Methode gefunden werden kann, wie aus vorliegenden Informationen (teil-)automatisiert entsprechende Informationsmodelle abgeleitet, bzw. Modellkomponenten semantisch richtig in bestehende Modelle eingeordnet, werden können. Die Handhabung und somit auch die benötigte Zeit bis zum Erreichen eines durchgängigen System-Managements könnte damit für jeden Hersteller, Integrator, Anwender und somit für die gesamte Branche deutlich erleichtert bzw. verkürzt werden.

Die Entwicklung von automatisierungsspezifischen Modellen und Methoden zum System-Management ist kein Prozess, der in absehbarer Zukunft abgeschlossen sein wird. Der Grund dafür ist, dass immer wieder und immer mehr Technologien, die ursprünglich keinen oder

wenig Entwicklungsbezug zu den Belangen der Automation hatten, in diesen Bereich drängen. Dadurch und durch Paradigmenwechsel in der Automation – z.B. Losgröße Eins – steigen die Komplexität und vor allem der Anspruch an ein Management aller Automatisierungskomponenten. Für jede dieser „neuen" Technologien werden wiederum Informationsmodelle für das Management benötigt, die die Spezifika der Automation widerspiegeln.

Bisher offen geblieben ist auch, wie zeitnah dafür gesorgt werden kann, dass automatisierungstechnische Geräte in ein modernes System-Management mit einbezogen werden können. Dies gilt vor allem für solche Geräte, die sich schon in Betrieb befinden und deren Austausch in den nächsten Jahren eher ausgeschlossen ist. Die grundsätzlich möglichen Konzepte zur Verteilung von WBEM/CIM-Servern und auch Providern, die in diesem Fall eingesetzt werden können, wurden in der Arbeit dargestellt. Die Herausforderung ist an der Stelle die räumliche Nähe zu den Geräten herzustellen, denn ein (Industrial)-PC, der für eine Vielzahl von Geräten als Management-Proxy fungiert, stellt zwar eine mögliche Lösung dar, ob sie jedoch optimal ist sollte in Zweifel gezogen werden. An dieser Stelle muss im Rahmen weiterer Arbeiten untersucht werden, ob dedizierte in Hardware umgesetzte Management-Proxies ein geeigneter Weg sind, um diese Konzeptlücke zu schließen. Solche Management-Proxies bieten weiterhin das Potential Management-Funktionen bereitzustellen, die auf Basis des Funktionsumfangs der eigentlichen automatisierungstechnischen Geräte technisch nicht möglich wären, beispielsweise das Sperren einzelner Dienste bzw. Ports eines Gerätes.

Obwohl die im Rahmen dieser Arbeit gesteckten Ziele erreicht werden konnten, bleiben gerade in Bezug auf Themengebiete wie Industrie 4.0 und die Nutzung moderner drahtloser Kommunikationsmittel in der Automation noch einiger Bedarf für weitere Entwicklungen.

9 Literaturverzeichnis

[1] DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE, *Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik (IEC 60050-351:2006),* 2009.

[2] R. Zurawski, The industrial communication technology handbook, Boca Raton, Fla. [u.a.]: Taylor & Francis, 2005.

[3] L. Urbas, A. Krause und J. Ziegler, Process control systems engineering, Oldenbourg Industrieverl.: Oldenbourg Industrieverl., 2012.

[4] B. Vogel-Heuser, C. Diedrich, A. Fay und P. Göhner, „Anforderungen an das Software-Engineering in der Automatisierungstechnik," in *Software Engineering 2013 - Fachtagung des GI-Fachbereichs*, Aachen, 2013.

[5] S. Lüders, „Defizite in puncto Cyber-Sicherheit," *Computer&AUTOMATION,* p. 26ff, März 2006.

[6] K. Ahmat, „Ethernet topology discovery: A survey," *arXiv preprint arXiv:0907.3095,* 2009.

[7] R. Black, A. Donnelly und C. Fournet, „Ethernet topology discovery without network assistance," in *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, 2004.

[8] J. Case, M. Fedor, M. Schoffstall und J. Davin, *RFC1157 - Simple Network Management Protocol (SNMP),* IETF, 1990.

[9] M. Metter und R. Bucher, Industrial Ethernet in der Automatisierungstechnik : Planung Und Einsatz Von Ethernet-LAN-techniken, Wiley, 2012.

[10] ITU-T, *Enhanced Telecom Operations Map (eTOM) - Interim view of an interpreter's guide for eTOM and ITIL practitioners,* 2007.

[11] A. Hanna und S. Rance, „ITIL Glossar und Abkürzungen," 2007.

[12] OASIS, *Reference Model for Service Oriented Architecture 1.0,* OASIS, 2006.

[13] T. Hadlich, *Verwendung von Merkmalen im Engineering von Systemen,* Magdeburg: Dissertation - Fakultät für Elektrotechnik und Informationstechnik, 2015.

[14] International Electrotechnical Commission, *IEC 61987-10:2009 - Industrial-process measurement and control - Data structures and elements in process equipment catalogues - Part 10: List of Properties (LOPs) for Industrial-Process Measurement and Control for Electronic Data Exchange - Fundamentals,* IEC, 2009.

[15] *DIN 66277:2014-08: Informationstechnik - Automatische Identifikation und Datenerfassungsverfahren - ElektronischesTypenschild,* Beuth, 2014.

[16] M. Dehof, M. Tangermann und A. Lüder, *SecIE: Handbook of Network Security,* Mannheim,: SecIE - Security and Administration in Industrial Ethernet e.V., 2007.

[17] H.-G. Hegering, S. Abeck und B. Neumair, Integrated Management of Networked Systems: Concepts, Architectures, and Their Operational Application, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998.

[18] U. Black, Network management standards: SNMP, CMIP, TMN, MIBs, and object libraries, McGraw-Hill, 1995.

[19] W. Stallings, SNMP, SNMPv2, and CMIP: the practical guide to network-management standards, Addison-Wesley, 1993.

[20] G. Held, Managing TCP/IP networks: techniques, tools, and security considerations, Wiley, 2000.

[21] M. Rose, *RFC1418 - SNMP over OSI,* IETF, 1993.

[22] ITU-T, *X.701: Information technology – Open Systems Interconnection – Systems management overview,* 1997.

[23] J.-P. Martin-Flatin, Web Based Management of IP Networks & Systems, Wiley, 2002.

[24] P. Ray, Integrated Management from E-Business Perspectives: Concepts, Architectures and Methodologies, Kluwer Academic/Plenum Publishers, 2003.

[25] A. Keller, „CORBA-basiertes Enterprise Management: Interoperabilität und Managementinstrumentierung verteilter kooperativer Managementsysteme in heterogener Umgebung," München, 1998.

[26] L. Cottrell, „Network Monitoring Tools," 14 Dezember 2015. [Online]. Available: http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html. [Zugriff am 15 Dezember 2015].

[27] R. Droms, *RFC2131 - Dynamic Host Configuration Protocol,* IETF, 1997.

[28] W. Croft und J. Gilmore, *RFC951 - Bootstrap Protocol,* IETF, 1985.

[29] International Electrotechnical Commission, *IEC 61158-5-10:2010 - Digital data communications for measurement and control–Fieldbus for use in industrial control systems–Part 5: Application Layer Service definition,* 3rd edition Hrsg., IEC, 2010.

[30] International Electrotechnical Commission, *IEC 61158-6-10:2010 - Digital data communications for measurement and control–Fieldbus for use in industrial control systems–Part 6: Application layer protocol specification,* 3rd edition Hrsg., 2010.

[31] M. Rose und K. McCloghrie, *RFC1155 - Structure and identification of management information for TCP/IP-based internets,* IETF, 1990.

[32] K. McCloghrie und M. Rose, *RFC1156 - Management Information Base for network management of TCP/IP-based internets,* IETF, 1990.

[33] J. Case, K. McCloghrie, M. Rose und S. Waldbusser, *RFC 1901 - Introduction to Community-based SNMPv2,* IETF, 1996.

[34] J. Case, K. McCloghrie, M. Rose und S. Waldbusser, *RFC1905 - Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2),* IETF, 1996.

[35] J. Case, K. McCloghrie, M. Rose und S. Waldbusser, *RFC1906 - Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2),* IETF, 1996.

[36] J. Case, R. Mundy, D. Partain und B. Stewart, *RFC3410 - Introduction and Applicability Statements for Internet-Standard Management Framework,* IETF, 2002.

[37] D. Harrington, R. Presuhn und B. Wijnen, *RFC3411 - An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks,* IETF, 2002.

[38] J. Case, D. Harrington, R. Presuhn und B. Wijnen, *RFC3412 - Message Processing and Dispatching for the Simple Network Management Protocol (SNMP),* IETF, 2002.

[39] D. Levi, P. Meyer und B. Stewart, *RFC3413 - Simple Network Management Protocol (SNMP) Applications,* IETF, 2002.

[40] U. Blumenthal und B. Wijnen, *RFC3414 - User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3),* IETF, Hrsg., IETF, 2002.

[41] B. Wijnen, R. Presuhn und K. McCloghrie, *RFC3415 - View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP),* IETF, 2002.

[42] R. Presuhn, *RFC3416 - Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP),* IETF, 2002.

[43] R. Presuhn, *RFC3417 - Transport Mappings for the Simple Network Management Protocol (SNMP),* IETF, 2002.

[44] R. Presuhn, *RFC3418 - Management Information Base (MIB) for the Simple Network Management Protocol (SNMP),* IETF, 2002.

[45] M. Knizak, M. Kunes, M. Manninger und T. Sauter, „Applying Internet management standards to fieldbus systems," in *IEEE International Workshop on Factory Communication Systems*, Barcelona, 1997.

[46] M. Kunes und T. Sauter, „Fieldbus-internet connectivity: the SNMP approach," *IEEE Transactions on Industrial Electronics,* Bd. 48, Nr. 6, pp. 1248-1256, 2001.

[47] ITU-T, *X.200 : Information technology - Open Systems Interconnection - Basic Reference Model: The basic model,* 1994.

[48] International Electrotechnical Commission, *IEC 61158: Industrial communication networks - Fieldbus specifications,* IEC.

[49] International Electrotechnical Commission, *IEC 61784 - 2:2014 - Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3,* IEC, 2014.

[50] The Open Group, „Systems Management: Common Manageability Programming Interface (CMPI)," 12 2006. [Online]. Available: https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?publicationid=12078. [Zugriff am 31 05 2015].

[51] M. E. Brasher, „KonkretCMPI," 6 Jun 2008. [Online]. Available: http://konkretcmpi.org/KonkretCMPI.pdf. [Zugriff am 17 Mai 2015].

[52] M. E. Brasher und K. Schopmeyer, „CIMPLE: An Embeddable CIM Provider Engine," in *ManDevCon*, Santa Clara, 2006.

[53] *DSP0004: CIM Infrastructure Specification,* DMTF, 2012.

[54] International Organization for Standardization, *ISO 15745-1:2003 - Industrial automation systems and integration -- Open systems application integration framework -- Part 1: Generic reference description,* ISO, 2003.

[55] V. Schiffer, „The Common Industrial Protocol (CIP) and the Family of CIP Networks," OVDA, Ann Arbor, 2006.

[56] M. Wollschlaeger und R. Frenzel, „Handling Field Device Documentations throughout the Life Cycle of Automation Systems - Web-based Information Model and Access Methods," in *2006 IEEE International Conference on Industrial Informatics*, Singapur, 2006.

[57] BITKOM e.V., VDMA e.V. und ZVEI e.V., „Umsetzungsstrategie Industrie 4.0 Ergebnisbericht der Plattform Industrie 4.0," BITKOM e.V., Berlin, 2015.

[58] D. Alexander, W. Martin und T+H, „Projekt SMartA Abschlussbericht Förderkenzeichen KF 2077606LF2," TU Dresden, 2015.

[59] R. Lehmann, A. Dennert und M. Wollschlaeger, „Diagnosis, Alarms and their Management in integrated Automation Systems," in *IEEE 20th Conf. Emerging Technologies and Factory Automation (ETFA)*, Luxembourg, 2015.

[60] IEEE, *IEEE 802.1AB-2009 Standard for Local and Metropolitan Area Networks-- Station and Media Access Control Connectivity Discovery,* New York, 2009.

[61] International Electrotechnical Commission, *IEC 62541-9:2012 - OPC Unified Architecture - Part 9: Alarms and conditions,* 2012.

[62] *DSP0202: CIM Query Language Specification,* DMTF, 2007.

[63] P. Elford, „Network Management: Is SNMP the Total Solution," in *AUUG 1992*, Melbourne, 1992.

[64] R. Lehmann, R. Frenzel und M. Wollschlaeger, „Integriertes System- und Dienste-Management," *atp edition,* Bd. 3/2012, p. 50 ff., 2012.

[65] M. Wollschlaeger, R. Lehmann und A. Dennert, „Life-Cycle-bezogene Information in Industrie 4.0," *atp edition,* Bd. 5/2015, Nr. 05, p. 24 ff., 2015.

[66] A. Pilz, „"Policy-Maker": a toolkit for policy-based security management," in *Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP*, 2004.

[67] A. Pilz und J. Swoboda, „Network management information models," *Aeu-International Journal of Electronics and Communications,* Bd. 58, Nr. 3, pp. 165-171, 2004.

[68] ITU-T, *M.3010 : Principles for a telecommunications management network,* 2000.

[69] ITU-T, *X.700 : Management framework for Open Systems Interconnection (OSI) for CCITT applications,* 1992.

[70] ITU-T, *X.710 : Information technology - Open Systems Interconnection - Common Management Information service,* 1997.

[71] ITU-T, *X.722 : Information technology - Open Systems Interconnection - Structure of management information: Guidelines for the definition of managed objects,* 1992.

[72] ITU-T, *X.730 : Information technology - Open Systems Interconnection - Systems Management: Object management function,* 1992.

[73] ITU-T, *X.751 : Information technology - Open Systems Interconnection - Systems Management: Changeover function,* 1995.

[74] G. Pavlou, „Telecommunications Management Network: A Novel Approach Towards its Architecture and Realisation Through Object-Oriented Software Platforms," 1997.

[75] ITU-T, *X.712 : Information technology - Open Systems Interconnection - Common management information protocol: Protocol Implementation Conformance Statement (PICS) proforma,* 1992.

[76] ITU-T, *X.711 : Information technology - Open Systems Interconnection - Common Management Information Protocol: Specification,* 1997.

[77] ITU-T, *X.721 : Information technology - Open Systems Interconnection - Structure of management information: Definition of management information,* 1992.

[78] ITU-T, *X.720 : Information technology - Open Systems Interconnection - Structure of management information: Management information model,* 1992.

[79] ITU-T, *X.725 : Information technology - Open Systems Interconnection - Structure of management information: General Relationship Model,* 1995.

[80] C. Hobbs, A practical approach to WBEM/CIM management, Auerbach, 2004.

[81] K. McCarthy, G. Pavlou, S. Bhatti, J. N. D. Souza und D. Souza, *Exploiting the Power of OSI Management for the Control of SNMP-capable Resources Using Generic Application Level Gateways,* 1995.

[82] OpenLMI, „OpenLMI Project," 05 2015. [Online]. Available: http://www.openlmi.org/. [Zugriff am 22 05 2015].

[83] DMTF, „DMTF Member List," 05 2015. [Online]. Available: http://dmtf.org/about/list. [Zugriff am 22 05 2015].

[84] C. Stögerer und W. Kastner, „Approaches for increasing availability of component-based traffic management software," in *Proc. 13th Int Intelligent Transportation Systems (ITSC) IEEE Conf*, 2010.

[85] L. So-Jung, C. Mi-Jung , Y. Sun-Mi , J. W. Hong, H.-N. Cho, C.-W. Ahn und . S.-I. Jung, „Design of a WBEM-based Management System for Ubiquitous Computing Servers," 28 07 2004. [Online]. Available: http://dmtf.org/sites/default/files/design%20of%20wbem%20based%20system.pdf. [Zugriff am 31 05 2015].

[86] DMTF, „TelcoWGCharter," 06 2007. [Online]. Available: http://dmtf.org/sites/default/files/TelcoWGCharter.pdf. [Zugriff am 31 05 2015].

[87] DMTF, „Management Profiles," DMTF, [Online]. Available: http://www.dmtf.org/standards/profiles. [Zugriff am 31 05 2015].

[88] O. Frommel, „Standards fürs Netzwerk- und System-Management," *ADMIN - IT-Praxis und Strategie,* Nr. 1, 2014.

[89] W. Mahnke, S.-H. Leitner und M. Damm, OPC Unified Architecture, Springer, 2009.

[90] OPC Foundation and PLCopen, *PLCopen and OPC Foundation: OPC UA Information Model for IEC 61131-3,* 3rd edition Hrsg., 2010.

[91] S. Rohjans, K. Piech, M. Uslar und J.-F. Cabadi, „CIMbaT -- Automated Generation of CIM--based OPC UA--Address Spaces," in *2011 Second IEEE International Conference on Smart Grid Communications*, 2011.

[92] S. Rohjans, M. Uslar und H. Appelrath, „OPC UA and CIM: Semantics for the smart grid," in *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, 2010.

[93] A. Fernbach, W. Granzer und W. Kastner, „Interoperability at the management level of building automation systems: A

case study for BACnet and OPC UA," in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, 2011.

[94] Java Community Process, *JSR 3 JavaTM Management Extensions (JMX)(TM)Specification, version 1.4 Maintenance Release 4,* 2013.

[95] Java Community Process, *JSR 160 JavaTM Management Extensions (JMX)(TM) Remote API Maintenance Release 2,* 2013.

[96] J.-P. Martin-Flatin, L. Bovet und J. P. Hubaux, „JAMAP: a Web-Based Management Platform for IP Networks," 1999.

[97] S. Joshi, I. Kazi, S. Shaikh, S. Sharma und P. Zhou, „Process-based cmpi provider management". USA Patent US20120311610 A1, 6 Dec 2012.

[98] DMTF, „CIM Schema: Version 2.36.0," DMTF, [Online]. Available: http://dmtf.org/standards/cim/cim_schema_v2360. [Zugriff am 5 Jun 2015].

[99] TMForum, „FRAMEWORX," TMForum, 2014. [Online]. Available: https://www.tmforum.org/tm-forum-frameworx/. [Zugriff am 13 Jannuar 2014].

[100] J. C. Strassner, Directory Enabled Networking, MacMillan Technical Publishing, 1999.

[101] International Electrotechnical Commission, *IEC 62453-2:2009 -Field device tool (FDT) interface specification - Part 2: Concepts and detailed description,* IEC, 2009.

[102] International Electrotechnical Commission, *IEC 62541-100:2015 - OPC Unified Architecture Specification - Part 100: Device Interface,* 1 Hrsg., IEC, 2015.

[103] International Electrotechnical Commission, *IEC62769-1:2015 - Field Device Integration (FDI) - Part 1: Overview,* 3rd edition Hrsg., IEC, 2015.

[104] Object Management Group, *ISO/IEC 19500-3:2012 Information technology - Object Request Broker Architecture (CORBA), Components,* OMG, 2012.

[105] Honeywell Inc. International, „OPC Server for SNMP," Juni 2014. [Online]. Available: http://www.matrikonopc.de/opc-server/opc-server-snmp.aspx. [Zugriff am Juni 2014].

[106] International Electrotechnical Commission, *IEC TR 62541-1:2010 - OPC Unified Architecture - Part 1: Overview and Concepts,* 2010.

[107] International Electrotechnical Commission, *IEC TR 62541-2:2010 - OPC Unified Architecture - Part 2: Security Model,* 2010.

[108] International Electrotechnical Commission, *IEC 62541-3:2010 - OPC Unified Architecture - Part 3: Address Space Model,* 2011.

[109] International Electrotechnical Commission, *IEC 62541-5:2011 - OPC Unified Architecture - Part 5: Information Model,* 2011.

[110] International Electrotechnical Commission, *IEC 62541-4:2011 - OPC Unified Architecture - Part 4: Services,* 2011.

[111] International Electrotechnical Commission, *IEC 62541-6:2011 - OPC Unified Architecture - Part 6: Mapping,* 2011.

[112] International Electrotechnical Commission, *IEC 62541-7:2012 - OPC Unified Architecture - Part 7: Profiles,* 2012.

[113] Kepware Technologies, Inc, „SNMP OPC Server," Juni 2014. [Online]. Available: http://www.kepware.com/Products/products_iSNMP.asp. [Zugriff am Jun 2014].

[114] Siemens AG, „SNMP OPC serverP," Juni 2014. [Online]. Available: http://w3.siemens.com/mcms/industrial-communication/en/ie/network-management/snmp-opc-server/Pages/snmp-opc-server.aspx. [Zugriff am Juni 2014].

[115] Obermeier-Software, „SNMP-OPC Server," Juni 2014. [Online]. Available: http://www.snmp-opc-gateway.com/. [Zugriff am Juni 2014].

[116] Softing AG Segment Industrial Automation, „SNMP-OPC-Server," Juni 2014. [Online]. Available: http://industrial.softing.com/de/produkte/funktionalitaet/sps-konnektivitaet/opc-server/snmp/opc-server-mit-snmp-

protokollunterstuetzung-einschliesslich-snmp-browse-und-mib-import-fuer-ethernet-tcpip.html. [Zugriff am Juni 2014].

[117] *VDI/VDE 2651 Blatt 1 Plant Asset Management (PAM) in der Prozessindustrie - Definition, Modell, Aufgabe, Nutzen,* VDI/VDE, 2009.

[118] M. Boucadair und C. Jacquenet, *RFC7149 - Software-Defined Networking: A Perspective from within a Service Provider Environment,* IETF, 2014.

[119] J. Case, K. McCloghrie, M. Rose und S. Waldbusser, *RFC1441 - Introduction to version 2 of the Internet-standard Network Management Framework,* IETF, 1993.

[120] J. Davin, J. Galvin und K. McCloghrie, *RFC1351 - SNMP Administrative Model,* IETF, 1992.

[121] *DSP0004:Common Information Model (CIM) Metamodel,* Bd. 3.0.0a, DMTF, 2013.

[122] *DSP0200: CIM Operations over HTTP,* DMTF, 2009.

[123] *DSP0201: Representation of CIM in XML,* DMTF, 2009.

[124] *DSP0205: WBEM Discovery Using the Service Location Protocol (SLP),* DMTF, 2014.

[125] *DSP0206: WBEM SLP Template,* DMTF, 2010.

[126] *DSP0207: WBEM URI Mapping,* DMTF, 2013.

[127] *DSP0210: CIM-RS Protocol,* DMTF, 2014.

[128] *DSP0211: CIM-RS Payload Representation on JSON,* DMTF, 2014.

[129] *DSP0226: Web Services for Management (WS Management),* DMTF, 2010.

[130] *DSP0223: Generic Operations Specification,* DMTF, 2013.

[131] *DSP0227: WS-Management CIM Binding Specification,* DMTF, 2010.

[132] DMTF, „The Value of the Common Information Model (Why CIM?),“ DMTF, 2003.

[133] R. Enns, M. Bjorklund, J. Schoenwaelder und A. Bierman, *RFC6241 - Network Configuration Protocol (NETCONF),* IETF, 2011.

[134] V. Fajardo, J. Arkko, J. Loughney und G. Zorn, *RFC6733 - Diameter Base Protocol,* IETF, 2012.

[135] R. Frenzel, R. Lehmann und M. Wollschlaeger, „Handling identification and maintenance information of intelligent field devices using Web Based Enterprise Management," in *Proc. IEEE 16th Conf. Emerging Technologies & Factory Automation (ETFA)*, Toulouse, 2011.

[136] R. Frye, D. Levi, S. Routhier und B. Wijnen, *RFC3584 - Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework,* IETF, 2003.

[137] J. Galvin und K. McCloghrie, *RFC1445 - Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)},* IETF, 1993.

[138] J. Galvin und K. McCloghrie, *RFC1446 - Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2),* IETF, 1993.

[139] J. Galvin, K. McCloghrie und J. Davin, *RFC1351 - SNMP Security Protocols,* IETF, 1992.

[140] P. Goncalves, J. Oliveira und R. Aguiar, „An evaluation of network management protocols," in *International Symposium on Integrated Network Management, 2009. IM '09. IFIP/IEEE*, Long Island, 2009.

[141] W. Granzer und W. Kastner, „Information modeling in heterogeneous Building Automation Systems," in *9th IEEE International Workshop on Factory Communication Systems (WFCS), 2012* , Lemgo, 2012.

[142] M. Hutter, A. Szekely und J. Wolkerstorfer, „Embedded system management using WBEM," in *International Symposium on Integrated Network Management, 2009. IM '09. IFIP/IEEE* , Long Island, 2009.

[143] J. Lange, F. Iwanitz und T. J. Burke, OPC von Data Access bis Unified Architecture, 4., völlig neu bearb. und erw. Aufl. Hrsg., VDE Verlag, 2010.

[144] D. Levi und J. Schoenwaelder, *RFC3165 - Definitions of Managed Objects for the Delegation of Management Scripts,* IETF, 2001.

[145] D. Levi, P. Meyer und B. Stewart, *RFC2273 - SNMPv3 Applications,* IETF, 1998.

[146] J. P. Martin-Flatin, „Push vs. pull in Web-based network management," in *6th IFIP/IEEE International Symposium on Integrated Network Management, 1999. Distributed Management for the Networked Millennium.* , Boston, 1999.

[147] J.-P. Martin-Flatin, „Web-Based Management of IP Networks and Systems," École Polytechnique Fédérale de Lausanne, Lausanne, 2000.

[148] S. Mätzler, M. Wollschlaeger, A. Fernbach, W. Kastner und M. Huschke, „An OPC UA cross-domain information model for energy management in automation systems," in *39th Annual Conference of the IEEE Industrial Electronics Society, IECON*, Wien, 2013.

[149] K. McCloghrie, *RFC1909 - An Administrative Infrastructure for SNMPv2,* IETF, 1996.

[150] K. McCloghrie, J. Davin und J. Galvin, *RFC1353 - Definitions of Managed Objects for Administration of SNMP Parties,* IETF, 1992.

[151] K. McCloghrie und J. Galvin, *RFC1447 - Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2),* IETF, 1993.

[152] K. McCloghrie, D. Perkins und J. Schoenwaelder, *RFC2578 - Structure of Management Information Version 2 (SMIv2),* IETF, 1999.

[153] M. Mishra und S. S. Bedi, „Web based enterprise management for distributed heterogeneous computing environment: A review," in *2014 Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH)*, 2014.

[154] M. Papazoglou und W.-J. van den Heuvel, „Web services management: a survey," *Internet Computing, IEEE,* pp. 58-64, Nov.-Dez. 2005.

[155] C. Rigney, S. Willens, A. Rubens und W. Simpson, *RFC2865 - Remote Authentication Dial In User Service (RADIUS),* IETF, 2000.

[156] J. Schneider, T. Kamiya, D. Peintner und R. Kyusakov, *Efficient XML Interchange (EXI) Format 1.0 (Second Edition),* 2014.

[157] M. Slavitch, *RFC2325 - Definitions of Managed Objects for Drip-Type Heated Beverage Hardware Devices using SMIv2,* IETF, 1998.

[158] C. Stogerer und W. Kastner, „System management standards for traffic management systems," in *IEEE Conference on Emerging Technologies Factory Automation, ETFA 2009.* , Malorca, 2009.

[159] C. Stögerer und W. Kastner, „Distributed monitoring for component-based traffic management systems," in *IEEE Conf. Emerging Technologies and Factory Automation (ETFA)*, Bilbao, 2010.

[160] D. van der Linden, W. Granzer und W. Kastner, „OPC Unified Architecture (OPC UA) new opportunities of system integration and information modelling in automation systems," in *9th IEEE International Conference on Industrial Informatics (INDIN)*, Lisabon, 2011.

[161] J. Veizades, E. Guttman, C. Perkins und S. Kaplan, *RFC2165 - Service Location Protocol,* IETF, 1997.

[162] S. Waldbusser, *RFC4502 - Remote Network Monitoring Management Information Base Version 2,* IETF, 2006.

[163] S. Waldbusser, *RFC1757 - Remote Network Monitoring Management Information Base,* IETF, 1995.

[164] U. Warrier, L. Besaw, L. LaBarre und B. Handspicker, *RFC1189 - Common Management Information Services and Protocols for the Internet (CMOT and CMIP),* IETF, 1990.

[165] G. Waters, *RFC1910 - User-based Security Model for SNMPv2,* IETF, 1996.

[166] *DSP1054: Indications Profile,* DMTF, 2011.

[167] *DSP1002: Diagnostics Profile,* DMTF, 2010.

[168] *DSP0231: CIM Simplified Policy Language (CIM-SPL),* DMTF, 2009.

[169] *DSP0230: WS-CIM Mapping Specification,* DMTF, 2011.

[170] D. Großmann, M. Braun, B. Danzer und M. Riedl, FDI - Field Device Integration, VDE Verlag, 2013.