

IDA – Interoperable, semantische Datenfusion zur automatisierten Bereitstellung von sichtenbasierten Prozessführungsbildern

M. Schleipen, Fraunhofer IOSB, Karlsruhe;
Dr. **M. Okon**, Fraunhofer IOSB, Karlsruhe;
T. Enzmann, cjt Systemsoftware AG, Stutensee;
Dr. **J. Wei**, cjt Systemsoftware AG, Stutensee;

Kurzfassung

Vor der Nutzung eines Leitsystems im realen Betrieb muss dieses projektiert werden. Der Projektierer erhält seine Informationen - wie Hallenlayout oder Signallisten - aus den verschiedenen Planungsphasen und verarbeitet diese größtenteils manuell. In diesem Beitrag wird das Projekt IDA vorgestellt, das sich mit der (teil-)automatisierten Bereitstellung von sichtenbasierten Prozessführungsbildern aus bereits vorhandenen Informationen befasst. Dieser Beitrag gibt einen Überblick über die allgemeine Architektur und hebt einige Aspekte der entwickelten Methoden und Werkzeuge (Tools) hervor. Ziel ist es, sowohl den Engineeringprozess effizienter zu gestalten als auch die Prozesse während des darauffolgenden Betriebs zu vereinfachen und zu verbessern.

Abstract (optional)

A production monitoring and control system has to be engineered before it can be used. For this task, the engineer receives information like hall layouts or signal lists from different preceding planning phases and processes them mostly manually. This contribution describes the project IDA which deals with (semi-)automatic generation of process images from different existing information sources. The contribution gives an overview about the general architecture and highlights some of the developed methods and tools. Goals are a more efficient engineering in the planning process and the optimization and simplification of processes during the operation phase.

1. Einleitung

Die Prozessvisualisierung ist ein wichtiger Teil der interaktiven – und damit sichtbaren - Softwarekomponenten eines Leitsystems für die Fertigungstechnik. Diese Softwarekomponente visualisiert den beobachteten und kontrollierten Produktionsprozess

(wie z.B. in Bild 1). Jedes Bild zeigt dabei einen anderen Produktionsschritt oder eine spezielle Sicht auf die Produktionsdaten. Sie dient dabei aber nicht nur zur Beobachtung des Produktionsprozesses sondern ermöglicht auch dessen manuelle Beeinflussung. Sie verbindet damit Mensch und Maschine und erlangt somit hohe Bedeutung. Prozessführungsbilder bieten unter anderem folgende wesentlichen Funktionen:

- Darstellung des technologischen Schemas der Produktionsanlagen und -prozesse
- Statische und dynamische Darstellung der Werte von Prozesssignalen, Produktionsdaten und weiteren Kontrollvariablen
- Manipulation der Daten
- Darstellung und Bestätigung von Alarmen
- Darstellung und Archivierung von Messwerten
- Umschaltmöglichkeiten zwischen verschiedenen Darstellungsarten oder Bildern

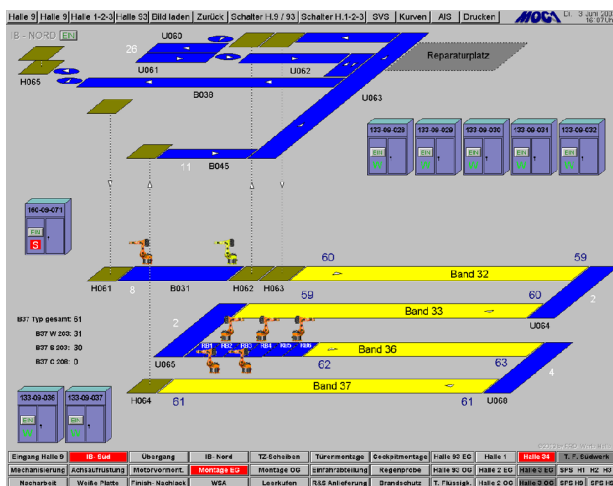


Bild 1: Prozessführungsbild eines Leitsystems

1.1 Motivation

Das Engineering von Leitsystemen umfasst die Herstellung der Verbindung zwischen der Ebene der Automatisierungskomponenten (z.B. SPSen) und dem eigentlichen Leitsystem. Die Schnittstelle zwischen Leitsystem und Bediener ist dabei die Leitsystem-Visualisierung. Normalerweise wird das Engineering der Leitsystem-Visualisierung von einem dafür zuständigen Ingenieur in mehreren Schritten durchgeführt:

- Er modelliert den Produktionsprozess unter Einbeziehung von Anlagen- und Zellenlayout, Materialfluss und Steuerungen. Anlagen und Steuerungen werden hier durch grafische Objekte oder mittels Ein-/Ausgabe Feldern (E/A) repräsentiert. In dieser Phase werden Typen und Anzahl der Bilder sowie die Navigation durch die Bildhierarchie bestimmt.

- Er verbindet die realen Prozesssignale mit spezifischen Elementen der konfigurierten Anlage und der Steuerung. In dieser Phase werden ausgesuchte Objekte der Prozessbilder dynamisiert. Diese können dann ihr Aussehen in Abhängigkeit von Prozesswerten oder Benutzeraktionen entsprechend verändern.

Das Ganze passiert hauptsächlich manuell und ist damit zeit- und kostenintensiv und überaus fehleranfällig. Das Einbringen von Prozessinformationen hängt hierbei entscheidend von den Fähigkeiten und Vorlieben des beteiligten Ingenieurs ab – es existiert nämlich bis heute keine Standardisierung hierfür in der diskreten Fertigungsindustrie.

1.2 Ziele

Das Projekt versucht unterschiedliche Ziele zu erreichen:

- Die Prozessvisualisierung soll automatisiert generiert werden, dabei werden bereits vorhandene Informationen mittels semantischer Relationen geeignet verknüpft.
- Der manuelle und damit fehleranfällige Teil der Arbeit soll verringert werden.
- Die Qualität der Lösung soll gesteigert werden. Dies soll mit der Bereitstellung verschiedener Sichten auf das System und die Anwendung erfolgen.
- Die Arbeit des Anlagenbedieners soll durch zusätzliche Visualisierungsmöglichkeiten vereinfacht werden.
- Fehler sollen durch ergonomischere Visualisierungen vermieden werden.

2. AutomationML als integriertes Datenaustauschformat

AutomationML [1], [2], [3] ist ein offenes, unabhängiges, XML-basiertes Datenformat zum Austausch von Anlagenplanungsdaten. Es versucht die Lücke zwischen Produktentwicklung und Produktion zu schließen. Es zielt darauf ab, die Interoperabilität zwischen produktionsrelevanten IT-Werkzeugen zu erhöhen, und kann in jeder Phase des Engineeringprozesses eingesetzt werden

Die allgemeine Architektur ist in Bild 2 dargestellt. Die Topologie von Objekten und deren Beziehungen untereinander werden dabei mit Hilfe des Formats CAEX (Computer Aided Engineering Exchange) [4] beschrieben. Zur Unterstützung von Geometrien und Kinematiken integriert AutomationML zusätzlich das Format COLLADA (COLLABorative Design Activity) [5], Logik und Verhalten wird mittels PLCOpenXML [6] beschrieben. Darüber hinaus lässt es die Integration weiterer relevanter Inhalte mittels XML-basierter Datenaustauschformate zu. Das z.Zt. gültige Modell ist vollständig skalierbar. Für die Ziele von IDA sind das Top-Level-Format CAEX und die Geometriebeschreibung mittels COLLADA von besonderem Interesse.

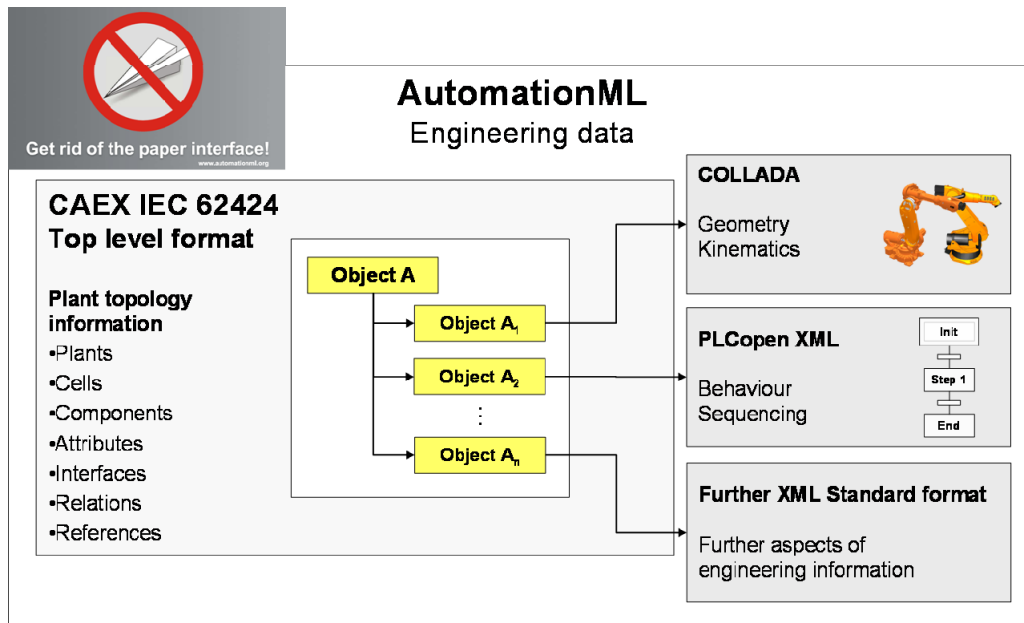


Bild 2: AutomationML Top-Level-Architektur [2]

2.1 CAEX

CAEX [4], [7] wurde in Zusammenarbeit des Lehrstuhls für Prozessleittechnik der RWTH Aachen und dem ABB Forschungszentrum in Ladenburg entwickelt und ist mittlerweile als IEC62424-Standard definiert. CAEX ist eine semiformale Datenbeschreibungssprache und basiert auf XML. Es beinhaltet ein Metamodell zur Beschreibung von Aufbau und Struktur von Anlagendaten. In erster Linie unterstützt das Format Bibliothekskonzepte und objektorientierte Ansätze. Es ermöglicht die Integration von Nutzer-, Anbieter- und Projektbibliotheken. Darüber hinaus werden Bottom-Up- und Top-Down-Entwurfsmöglichkeiten unterstützt. Die technische Innovation dieses Ansatzes ist die syntaktische und semantische Vereinheitlichung von Daten. Dies erlaubt die Entkopplung der erforderlichen Konfigurationsalgorithmen von den eigentlichen Datenquellen. [8]

CAEX besteht aus drei Bibliothekstypen: InterfaceClassLibraries, RoleClassLibraries und SystemUnitClassLibraries. Zusätzlich gibt es eine InstanceHierarchy, in der die konkrete Anlage modelliert ist. Desweiteren gibt es Elemente zur Detailspezifikation oder Verbindungen (Links) zwischen Elementen.

2.2 COLLADA

COLLADA ist eine Austauschmöglichkeit für interaktive dreidimensionale Inhalte. Es beschreibt die geometrische Gestalt eines Objektes. Ein Vorteil von COLLADA sind präzise Objektbeschreibungen durch Netze (Meshes). Darüber hinaus gibt es weitere Typen von

Informationen, die integriert werden können: Information über Kinematik, Material und Struktur. Die Informationen in einer COLLADA-Datei sind für Mensch und Maschine gleichermaßen leicht lesbar, da auch dieses Format auf XML basiert. [5]

3. Allgemeine Architektur von IDA

Die allgemeine Architektur von IDA zerfällt in zwei Teile, den IDA-Fusionsassistenten und den IDA-Bildassistenten.

3.1 Import und Fusion

Import und Fusion der Daten erfolgt im IDA-Fusionsassistenten (siehe Bild 3), dabei können verschiedene mögliche Datenformate importiert werden. CAEX beschreibt die Topologie und die E/A-Verbindungen. Dies kann z.B. durch den im Projekt ProduFlexil [9],[10] entwickelten E/A-Importer geschehen. Die 2D-Topologie und -Geometrie kann durch einen CAEX Importer behandelt werden. Der COLLADA-Importer kümmert sich um 3D-Geometrien und -Layouts. Ein weiteres Datenformat für Hallenlayouts ist das Drawing Interchange Format (DXF, [11]). DXF ist ein ASCII-basierter Standard, der von Autodesk spezifiziert wurde. Eine DXF-Datei beschreibt prinzipiell ein CAEX-Modell als Vektor-Grafik. Unterstützte Geometrieelemente sind Punkt, Linie, Kreis, Spline, etc. Dieses Format wird durch einen zusätzlichen DXFToCollada Konverter unterstützt. Am Schluss integriert ein Datenfusionierer alle Informationen zu einer gemeinsamen Beschreibung.

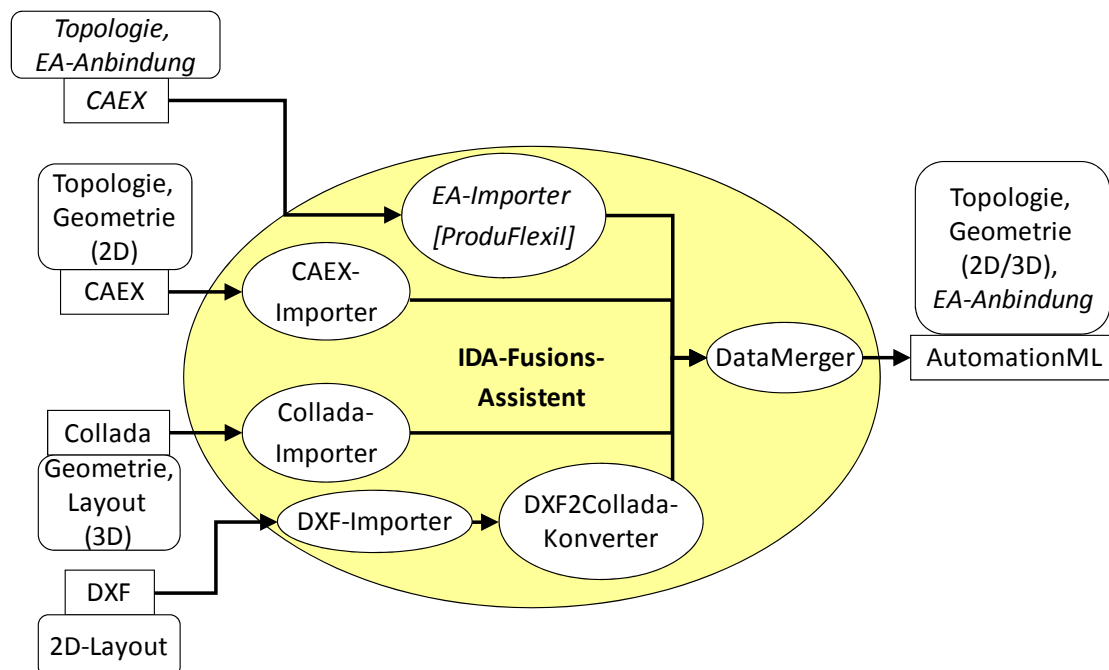


Bild 3: IDA-Fusionsschritt

Die Daten sind in verschiedene Aspekte unterteilt. Topologie und E/A werden in CAEX, 3D-Geometrien in COLLADA beschrieben. Die 2D-Daten eines Hallenlayouts dienen zur Positionierung der spezifischen 3D-Geometrien. Dies alles ist Teil eines AutomationML-Modells, das wiederum als Basis für die Generierung der Prozessvisualisierung dient. In Bild 4 werden die Beziehungen zwischen einem 2D-Hallenlayout, einem AutomationML-Modell, der 3D-Detail-Information und der generierten Prozessvisualisierung dargestellt.

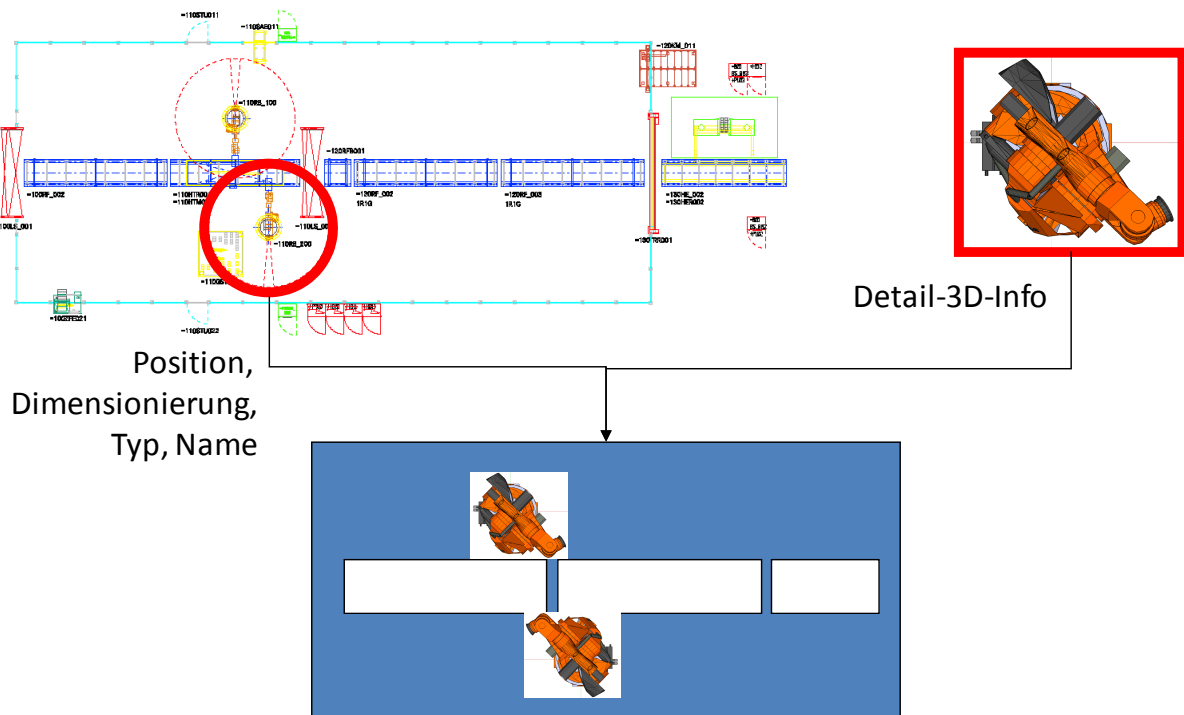


Bild 4: Fusion von 3D- und 2D-Information

Aus dem DXF-Layout werden Position und Orientierung der Anlagenkomponenten (hier ein Roboter) innerhalb der Produktionszelle, -linie oder -halle gewonnen. Position und Orientierung werden in einem AutomationML-CAEX-Frame-Attribut eingebettet, welches aus x, y und z als 3D-Punkt und rx, ry und rz als Roll-Pitch-Yaw-Winkeln besteht. Das Frame ist Teil eines AutomationML Objekts (eines Roboters) in der InstanceHierarchy. Mit demselben Objekt wird die COLLADA-3D-Geometrie durch ein ExternalInterface vom Typ ColladaRepresentationInterface verbunden. Die DXF-Daten können zusätzlich nach einer geeigneten Transformation als COLLADA-Geometrie-Daten verwendet werden. Eine komplette AutomationML-Szene besteht aus verschiedenen AutomationML-Objekten in der InstanceHierarchy, die alle von SystemUnitClasses abgeleitet sind, eine zu ihrer Semantik korrespondierende RoleClass definieren und weitere Interfaces besitzen, die sie mit anderen AutomationML Objekten verbinden.

In diesem Projekt besteht eine der Beispielapplikationen aus einer Transportlinie, bestehend aus sieben Transportbändern, vier Drehtischen und einer Teststation. Jedes Transportband ist ein InternalElement (AutomationML Objekt) in der InstanceHierarchy. Es ist bidirektional abgeleitet von einer SystemUnitClass Transportband und erfüllt die Rolle Transportband. Es hat verschiedene ExternalInterfaces, z.B. ColladaRepresentationInterface zur Zuordnung der 3D-Geometrie an das Objekt und die Order-Schnittstelle zur Beschreibung von Material- oder Prozessfluss. Mittels der Frame-Attribute der InternalElements kann also eine komplette Szene der Produktionszelle aufgebaut werden. Eine solche Szene zeigt Bild 5.

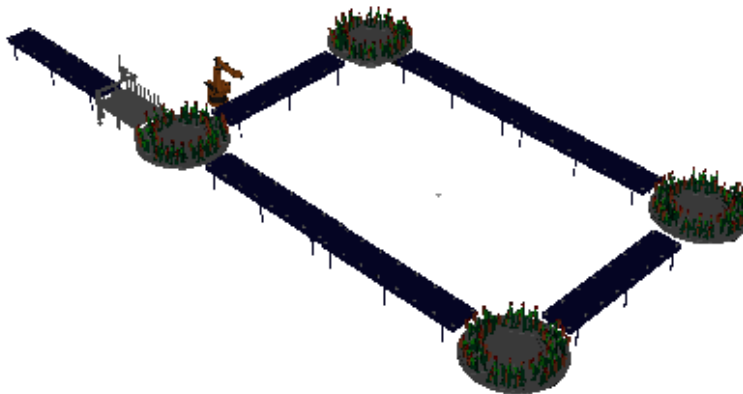


Bild 5: Beispielapplikation

3.2 Aufbereitung und Projektion

Nach Fusionierung aller relevanten Daten in ein konsistentes AutomationML-Modell müssen noch viele Probleme bis zur Darstellung von Objekten in einer Prozessvisualisierung gelöst werden.

Zunächst einmal unterstützen die meisten Prozessvisualisierungssysteme lediglich 2D-Grafiken. Weiterhin wären diese durch die detailtreuen CAD-Daten überfordert, konzentrieren sie sich doch nicht auf eine Einzeldarstellung, sondern bieten in einem Bild oft 50 oder mehr verschiedene Objekte an. Der Unterschied zwischen verschiedenen Detaillierungstiefen ist in Bild 6 (links) sichtbar.

Die Probleme beim Herunterbrechen von 3D- auf 2D-Daten sind hierbei vielfältig. Bild 6 (rechts) zeigt, dass, abhängig vom Betrachtungspunkt (hier: von oben), das entsprechende Objekt nicht eindeutig bestimmbar ist: Beide Objekte auf der linken Seite führen hier zur selben Ansicht.

Der Umriss eines Objekts ist problematisch. Er kann den Blick eines Nutzers auf ein Objekt leicht verfälschen. Ein zentraler Punkt in der Prozessvisualisierung ist aber die schnelle und korrekte Identifikation von Objekten durch den Benutzer. Damit ist das ein wichtiges Problem. Durch den Umriss eines Objekts ist es im Allgemeinen nicht möglich ohne

zusätzliche Informationen zu erkennen, um was für ein Objekt es sich jeweils handelt. Dagegen sind die sichtbaren Ecken reich an Information. Das Problem hier ist allerdings die Komplexität der Darstellung.

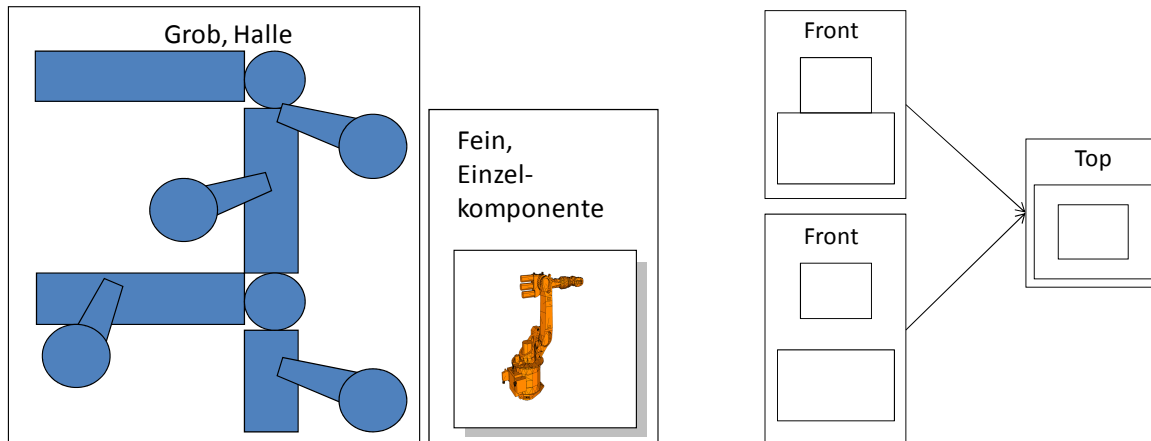


Bild 6: Verschiedene Detaillierungstiefen (links), Ansichten eines Objekts (rechts)

Für die Aufbereitung und Projektion der Geometrien – kurz: Darstellung – existieren daher verschiedene Stufen und damit einhergehende Komplexitätsgrade.

In Stufe 1 können für alle Objekte Bitmaps genutzt werden. Dabei werden Screenshots der dreidimensionalen Geometrien in verschiedenen Sichten (Oben, Vorne, Hinten, rechte Seite, linke Seite, Iso) angefertigt und als Bitmaps oder ähnliches gespeichert werden (siehe Bild 7). Diese Bitmaps können in die Prozessvisualisierung integriert werden. Der Nachteil dieser Methodik besteht in der Dynamisierung der Objekte. Ist beispielsweise der Status einer Anlagenkomponente an ein Objekt angebunden, blinkt dieses im Falle eines Fehlers in einer bestimmten Farbe (üblicherweise rot). Gibt es überlappende Objekte führt dies schnell zur Verwirrung. Eine mögliche Erweiterung oder Verbesserung besteht darin, Bitmaps für jedes Anlagenteil zu generieren. Dieses Konzept ist als ein On-Top-Plugin in einen bestehenden Layout-Manager eines Visualisierungssystems bereits integriert (siehe [12], [13]).

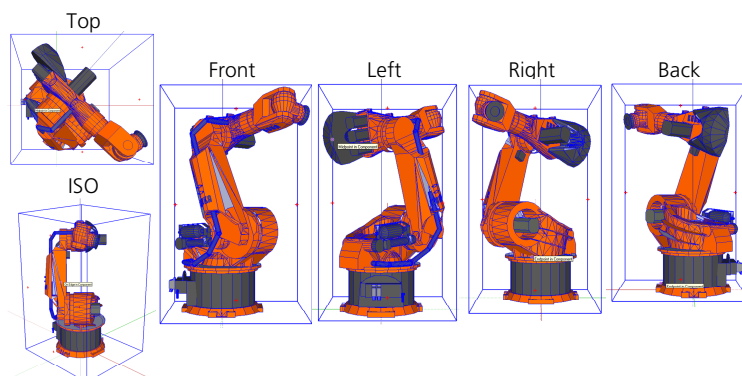


Bild 7: Ansichten aus verschiedenen Perspektiven

Der Ansatz in der Stufe 2 im Projekt IDA ist die Generierung einer Bitmap zusammen mit einer umgebenden Kontur. Bild 8 zeigt dies. Vorteile sind hier die einfache Struktur (Bitmap) des Objekts und den fehlenden Informationsverlust im Auge des Betrachters.

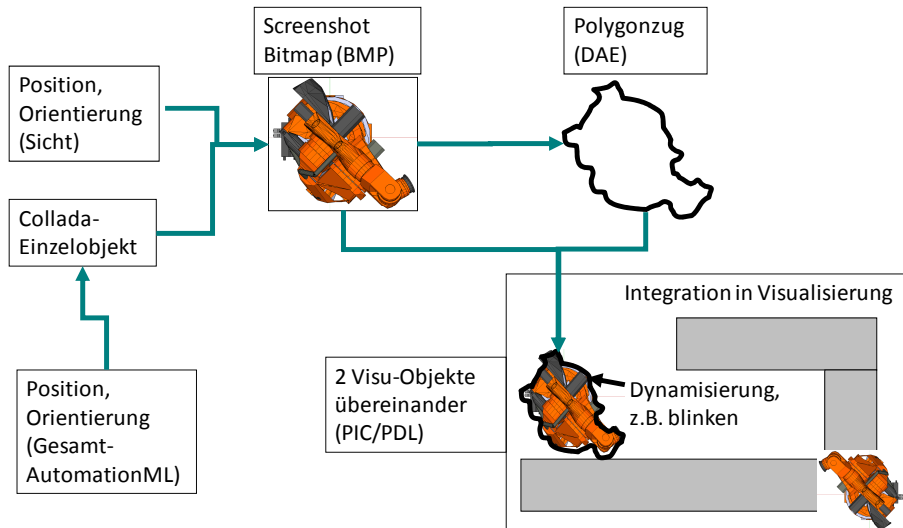


Bild 8: Kontur- und Bitmap-Fusion zur Darstellung im Bild

Der Ansatz der Stufe 3 behandelt komplexe Objekte (siehe Bild 9, links). Der erste Schritt ist die Projektion von 3D nach 2D. Im Projekt IDA geschieht dies durch eine Parallelprojektion (im Gegensatz zu einer Zentralprojektion oder einem Schnitt). Deshalb wird jeder Punkt in der COLLADA-Beschreibung auf die x-z-, y-z-, x-y- oder ISO-Ebene projiziert. Vorher werden noch die sichtbaren Konturen mittels einer geeigneten ‚Normalenkalkulation‘ der zugrunde liegenden Triangulation extrahiert: Zeigt die Normale eines Oberflächendreiecks in die Richtung der Betrachtungsposition (Kamera), so ist dieses Oberflächenteil sichtbar und wird zur Projektion herangezogen. Dies kann durch das Skalarprodukt der entsprechenden Normalenvektoren (> 0) leicht kalkuliert werden. Nach der Generierung dieser 2D-Daten im 3D-Raum (siehe Bild 9, rechts), muss die Komplexität weiter reduziert werden, da Prozessvisualisierungen in kurzer Zeit eine Menge solcher Objekte verarbeiten müssen.

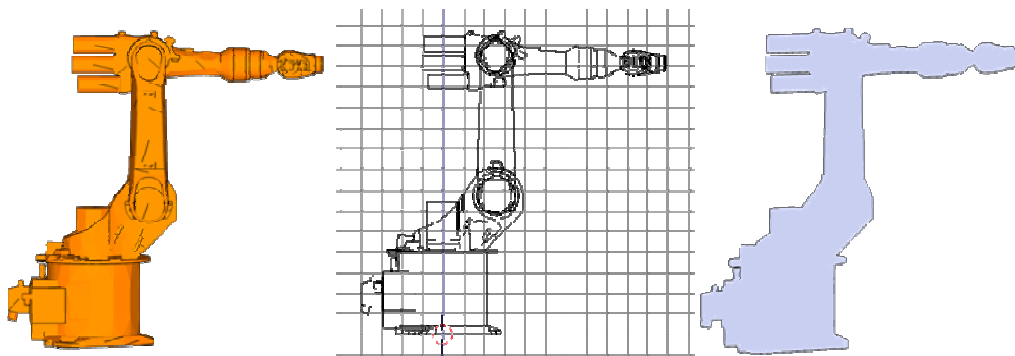


Bild 9: Reduktion der Informationsdichte

3.3 Sichtengenerierung

Die Sichtengenerierung geschieht in mehreren Schritten und ist eng verbunden mit dem Export (siehe Bild 10). Zusammen ergeben sie den IDA-Bildassistenten.

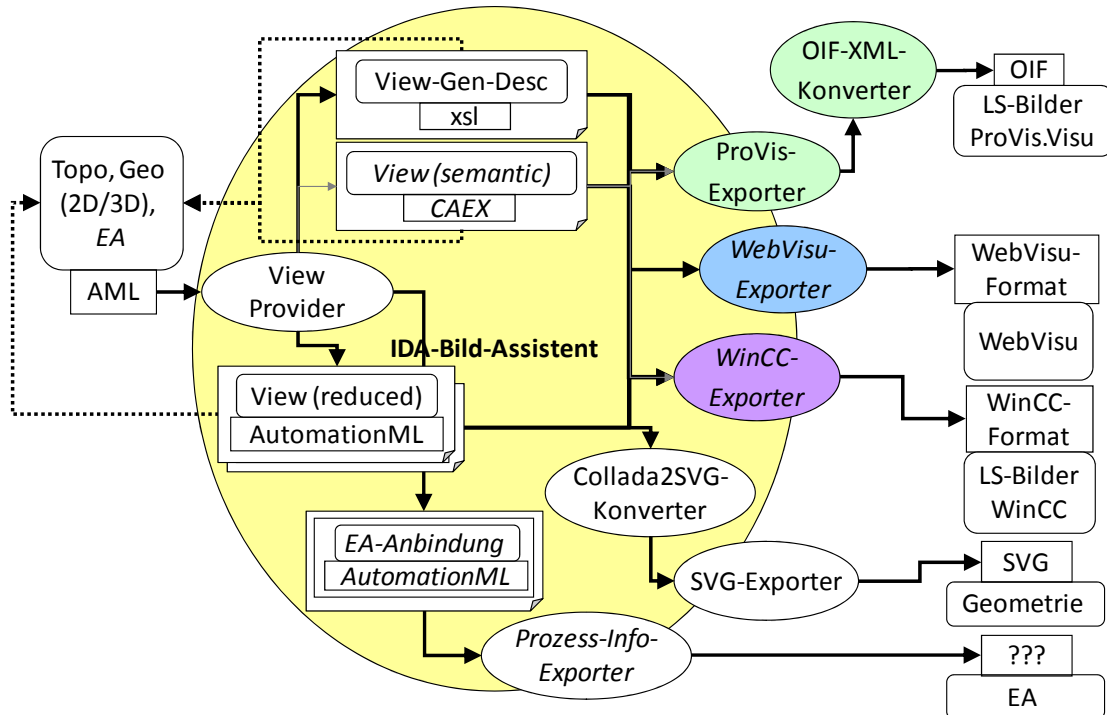


Bild 10: IDA-Bildassistent

Der Benutzer kann konfigurieren wie Sichten und Bilder generiert werden sollen. Er bestimmt Darstellungen und Bildaufbau und konfiguriert so seine persönliche Sicht. Bild 11 zeigt das Zusammenspiel der beteiligten Komponenten

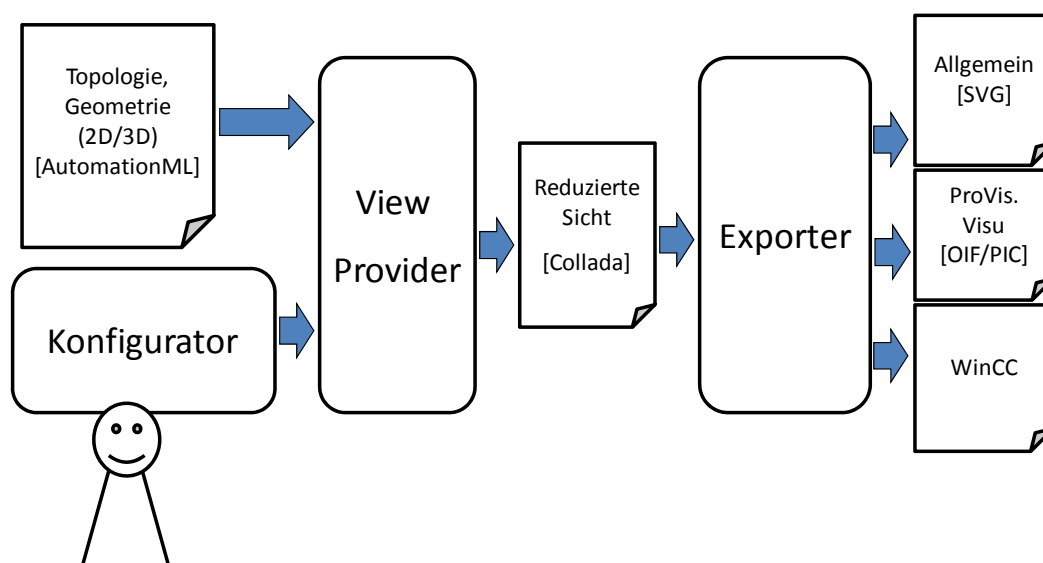


Bild 11: Sichtengenerierung

Der Benutzer erhält eine AutomationML-Datei mit Topologie- und Geometriedaten. Er definiert oder hat bereits seine persönliche Sicht definiert. Diese Information wird zusammen mit der AutomationML-Datei an den View-Provider weitergereicht. Der View-Provider generiert mit dieser Information oder Aktionslisten die reduzierten Ansichten. Die Konfiguration kann also als Anleitung zur Sichtengenerierung gesehen werden.

3.4 Export

Am Ende des Prozesses (siehe Bild 12) muss die reduzierte Sicht noch in das gewünschte Prozessvisualisierungssystem exportiert werden. Im Fall von ProVis.Visu® müssen dazu die Daten in das proprietäre XML-basierte OIF-Format umgewandelt werden. Diese Schnittstelle hilft beim Import und der Manipulation der systemspezifischen Daten. Für jedes Visualisierungssystem muss entsprechend ein spezifischer Importer entwickelt werden.

4. Zusammenfassung

In diesem Beitrag wurde die allgemeine Architektur des Forschungsprojektes IDA erläutert. Bild 12 beschreibt den gesamten Prozess mit allen entwickelten Methoden und Tools im Projekt IDA. Beginnend mit dem Import von Daten aus unterschiedlichen Quellen werden die bereit gestellten Daten fusioniert. Danach, die Daten sind entsprechend aufbereitet, werden Projektionen von 3D auf 2D bereitgestellt. Mit diesen Daten werden Sichten generiert und in die proprietären Prozessvisualisierungsformate exportiert. Alle Konzepte wurden mittels der erwähnten Beispielanwendung evaluiert. Im Bereich der Sichtengenerierung und der Aufbereitung/Projektion gibt es noch eine Menge offener Fragen und Forschungsfelder.

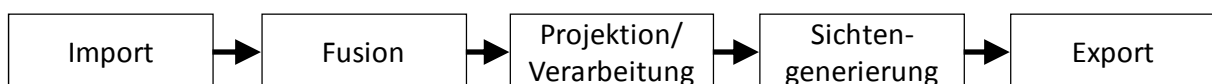


Bild 12: IDA-Ablauf

5. Danksagung

Das dem Beitrag zugrundeliegende Projekt wird gefördert vom Bundesministerium für Wirtschaft und Technologie aufgrund eines Beschlusses des Deutschen Bundestages unter dem Förderkennzeichen KF2074702ED9. Der Autor übernimmt die Verantwortung für den Inhalt des Beitrags.

Literatur

- [1] R. Drath et. al., Datenaustausch in der Automatisierungsplanung mit AutomationML. Springer Verlag, Berlin, 2009.

- [2] AutomationML, www.automationML.org, 2010.
- [3] M. Schleipen, R. Drath, "Three-View-Concept for modeling process or manufacturing plants with AutomationML", 13th IEEE ETFA 2009, September 2009.
- [4] M. Fedai, U. Epple, R. Drath, A. Fay, "A Metamodel for generic data exchange between various CAE Systems," In: Proceedings of 4th Mathmod Conference, Edition 24 of ARGESIM Report, ISBN 3-901608-24-9, pp. 1247-1256, 2003.
- [5] Remi Arnaud: Collada: Sailing the Gulf of 3D Digital Content Creation. 30.September 2006.
- [6] PLCopen. <http://www.plcopen.org/>, Stand 06.04.2011.
- [7] M. Fedai, R. Drath, "CAEX - a neutral data exchange format for engineering data," In: *atp/international 1/2005*, Oldenbourg Industrieverlag, pp. 43-51, 2005.
- [8] M. Schleipen, R. Drath, O. Sauer, "The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system", IEEE International Symposium on Industrial Electronics – ISIE 2008, Cambridge, pp. 1786-1791, 2008.
- [9] Miriam Schleipen, Michael Okon, Michael Baumann, Martin Neukaeufer, Christian Fedrowitz, Martin Feike, Nataliya Popova, Markus Nick, Soeren Schneickert, Martin Wessner: Design and engineering processes in highly adaptive plants with ambient intelligence techniques. Proceedings of 42nd CIRP Conference on Manufacturing Systems, 3.-5.6.09, Grenoble.
- [10] Miriam Ebel, Michael Okon, Michael Baumann: „ProduFlexil“: Flexible Produktion mit SOA-Architektur und Plug-and-Work-Mechanismus
Tagungsband zum Stuttgarter Softwaretechnik Forum (Science meets business), S.65-74, 20.-23. November 2007.
- [11] http://de.wikipedia.org/wiki/Drawing_Interchange_Format, Stand 06.04.2011.
- [12] M. Schleipen, K. Schick, "Self-configuring visualization of a production monitoring and control system", CIRP ICME 08, July 2008.
- [13] M. Schleipen, O. Sauer, "Usage of dynamic product and process information in a production monitoring and control system by means of CAEX and OPC UA", 3rd CIRP CAPE 2009, 2009, pp. 662-671.