

Labormodell zur Fabrikautomatisierung an der Otto-von-Guericke-Universität Magdeburg, Lehrstuhl für Fabrikbetrieb und Produktionssysteme

Bild: Uni Magdeburg

Integration von heterogenen Engineering-Daten mit AutomationML und dem AML.hub

## Konsistente Daten über Fachbereichsgrenzen hinweg

Im verteilten und parallelen Engineering industrieller Anlagen sind abgestimmte Engineering-Pläne und Daten erforderlich, um Fehler und Risiken frühzeitig zu identifizieren und zu beheben. Speziell im Umfeld von Industrie 4.0 sind ein effizienter Datenaustausch unterschiedlicher Fachbereiche und ein integriertes Anlagenmodell der fachbereichsübergreifenden Konzepte unumgänglich. Basierend auf dem standardisierten Datenaustauschformat AutomationML (AML) hilft der AML.hub dabei, Engineering-Daten über Fachbereichsgrenzen hinweg konsistent zu halten, Fehler zeitnah zu finden und damit Risiken zu minimieren sowie ein derartiges Anlagenmodell zu erstellen.

Im Anlagen-Engineering ist eine Vielzahl unterschiedlicher Fachbereiche wie beispielsweise Mechanik, Elektrotechnik, Prozessplanung oder Softwaretechnik an der Entwicklung eines heterogenen Automatisierungssystems beteiligt [7]. In den jeweiligen Fachbereichen verwenden Ingenieure meist bewährte Software-Werkzeuge mit spezifischen Datenmodellen, um Engineering-Pläne zu erstellen („Best-of-Breed“). Unterschiedliche Software-Werkzeuge und Datenmodelle sind allerdings oft nur unzureichend zuverlässig aufeinander abgestimmt und erschweren den Datenaustausch zwischen den Fachbereichen [3]. Das kann zu Fehlern und Inkonsistenzen führen und erhöht das Projektrisiko.

### Effizienter Datenaustausch

Durch technische und semantische Mechanismen für einen effizienten Datenaustausch [7] und die Synchronisierung von Engineering-Plänen können Projektdaten konsistent gehalten und Fehler effizient und frühzeitig erkannt werden. Betroffene Ingenieure können bei Änderungen oder Fehlern fachbereichsübergreifend benachrichtigt werden. Dazu sind allerdings auch angepasste Engineering-Prozesse erforderlich, die beispielsweise Round-Trip-Engineering und den Aufbau eines integrierten Anlagenmodells ermöglichen. Gemeinsame Konzepte [9] können diese semantische Lücke zwischen Datenmodellen aus Fachbereichen schließen und ermögli-

chen neben abgestimmten Daten auch den Einsatz von Abfragen über Fachbereichs- und Werkzeuggrenzen hinweg. Ein gemeinsames standardisiertes Datenaustauschformat, wie beispielsweise AutomationML (AML, [www.automationml.org](http://www.automationml.org)) sowie ein darauf aufbauendes Anlagenmodell können dabei helfen, unterschiedliche Sichten auf Engineering-Daten (etwa aus Perspektiven unterschiedlicher Fachbereiche) einheitlich darzustellen und zu verwenden [1][2][6]. Voraussetzung dafür ist die Verfügbarkeit geeigneter Schnittstellen von Software-Werkzeugen zur Bereitstellung der Engineering-Daten im Datenformat AML [7], wie etwa CAEX, Collada oder PLCopen XML.

Die Verwendung von AML kann die Integration von Datenmodellen unterschiedlicher Fachbereiche erleichtern, erfordert aber eine geeignete Werkzeugunterstützung für

- einen effizienten Datenabgleich,
- die Versionierung der Daten,
- das Herstellen von Verbindungen zwischen unterschiedlichen Plänen sowie
- die Bereitstellung passender Sichten auf Engineering-Daten.

**Abbildung 1** zeigt das Konzept eines ‚AML Repositories‘ für integrierte Daten sowie involvierte Software-Werkzeuge, deren Datenmodelle und beteiligte Ingenieure.

### Der AML.hub als Integrationsplattform

Der AML.hub, der vom Christian Doppler Forschungslabor CDL-Flex an der TU Wien gemeinsam mit Partnern entwickelt wurde, stellt eine Plattform dar, die einen effizienten und qualitätsgesicherten Datenaustausch zwischen unterschiedlichen Fachbereichen in einem heterogenen Entwicklungsumfeld, basierend auf dem AML-Datenformat [6], effizient unterstützt. Der konzeptionelle Prototyp des AML.hub (siehe **Abbildung 2**) beinhaltet:

- (1) *spezifische Sichten* auf das Projekt (Rollen bzw. Fachbereiche)
- (2) *Projektstruktur* auf Verzeichnis- und Dateibasis: Diese Elemente können sowohl strukturierte AML-Daten als auch Dateien anderer Formate beinhalten. Strukturierte AML-Daten werden analysiert und automatisch verknüpft. Liegt keine Strukturinformation vor (etwa Abbildungen oder unstrukturierter Text), steht die gesamte Datei als Referenz zur Verfügung
- (3) *integrierte Topologie* der gesamten Anlage
- (4) *Verknüpfungen* von Modellinformationen und AML-Funktionselementen
- (5) spezifische Sichten auf einzelne Anlagenelemente und deren Verknüpfungen (*Model-Link-View*), Versionierungen (*History View*) sowie Abfragen über AML-Modelle und Versionen (*Queries*)

### Prototypische Umsetzung des AML.hub

**Abbildung 3** beschreibt einen einfachen Ablauf zur Illustration der Funktionsweise des AML.hub in einem heterogenen Entwicklungsumfeld am Beispiel des ‚Labormodells zur Fabrikautomatisierung‘[8]: Der Anlagenplaner (*Plant Planner*) definiert die Anlagentopologie als Startpunkt für das Projekt und stellt sie als AML-Daten im AML.hub bereit (1). Die Daten können mittels Drag&Drop aus der lokalen Verzeichnisstruktur direkt in den ‚Project Explorer‘ übernommen werden und werden dort automatisch verarbeitet. Der Mechanik-Entwickler übernimmt die Topologie aus dem AML.hub, ergänzt spezifische Pläne aus dem Bereich Mechanik und stellt die geänderten Pläne bereit (3). Anschließend liest der Elektrik-Entwickler sowohl die Topologie als auch die mechanischen Pläne (4), soweit sie für ihn relevant sind, ergänzt diese entsprechend und stellt sie wiederum über den AML.hub zur Verfügung (5). Abschließend übernimmt der PLC-Programmierer seine fachbereichsspezifischen Daten aus dem AML.hub und ergänzt die relevanten Softwarekom-

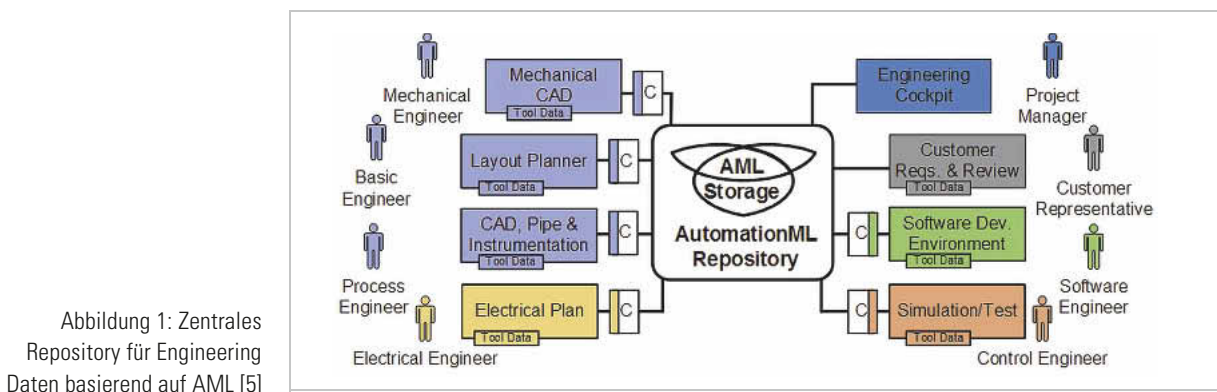


Abbildung 1: Zentrales Repository für Engineering Daten basierend auf AML [5]

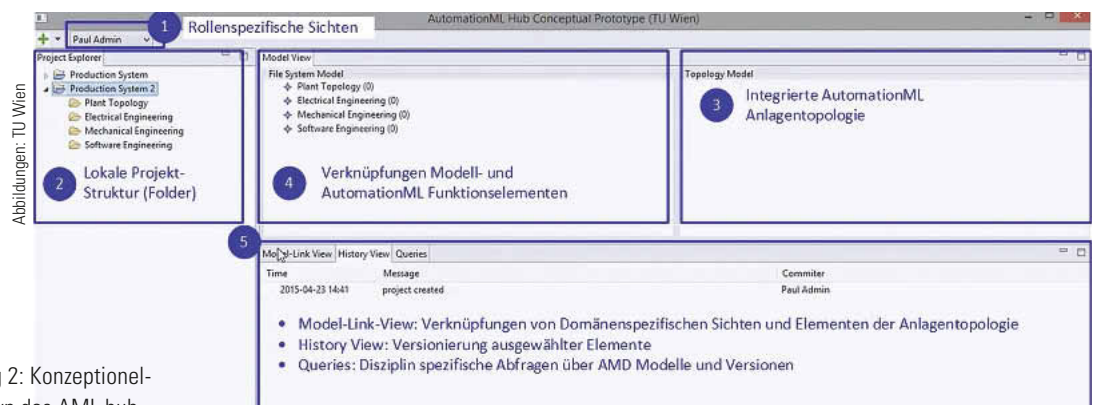


Abbildung 2: Konzeptioneller Prototyp des AML.hub

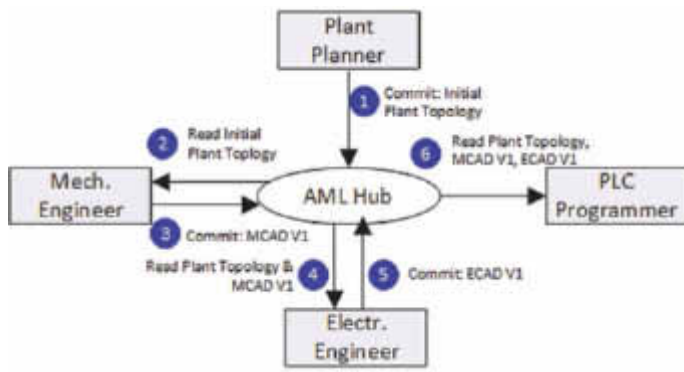


Abbildung 3: Einfache Werkzeugkette mit dem AML.hub

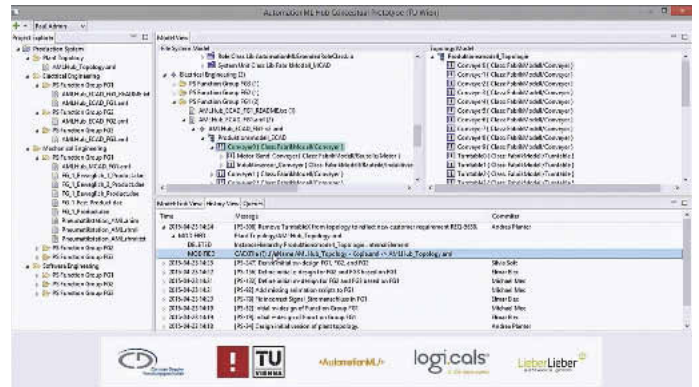


Abbildung 4: Screenshot aus der Prototypenanwendung des AML.hub

ponenten. **Abbildung 4** zeigt einen Screenshot des AML.hub Prototypen nach Abarbeitung dieser vereinfachten Werkzeugkette. In der Praxis arbeiten die einzelnen Fachbereiche parallel, so dass es beim Zusammenführen (*Checkin*) zu Konflikten und Überschneidungen kommen kann. Beispielsweise kann eine erforderliche Änderung in einem Arbeitsschritt notwendige Nacharbeiten (auch in anderen Fachbereichen) auslösen. Daher ist sowohl eine *Versionskontrolle* als auch eine Integration spezifischer Daten oder Versionen (*Merging*) erforderlich – Funktionalitäten, die der AML.hub bereitstellen kann.

**Schlussbemerkungen**

Die effiziente Integration heterogener Engineering-Daten ist ein erfolgskritischer Faktor für das Anlagen-Engineering, auch im Industrie-4.0-Kontext [4]. Semantische Lücken zwischen unterschiedlichen Fachbereichen können durch die Verwendung eines standardisierten Datenformates, wie AutomationML [1][2], und durch die Verwendung von Integrationsplattformen, wie etwa den AML.hub, geschlossen werden. Dadurch ergeben sich zahlreiche Vorteile:

- Fachexperten können weiterhin ihre bewährten Software-Werkzeuge verwenden.
- Ausgewählte Planungsdaten können beispielsweise an Lieferanten oder Projektpartner weitergegeben und qualitätsgesichert wieder zusammengeführt werden.
- Engineering-Daten (Topologie und Daten involvierter Fachbereiche) sind über Werkzeuggrenzen hinweg durch ein integriertes Anlagenmodell verknüpft und konsistent.
- Die einfache Versionierung von Planungsdaten erlaubt Analysen und die Integration von Engineering-Plänen einschließlich einer Konflikt- und Fehlererkennung.
- Analyse der Auswirkung von Änderungen basierend auf integrierten Daten.

Durch effiziente Integrationsmechanismen und die Verfügbarkeit von abgestimmten und konsistenten Daten können nicht nur Entwicklungsprozesse verbessert sondern auch Projektrisiken und Fehler reduziert werden. Weiterhin ermöglicht der AML.hub – aufbauend auf dem Datenformat AutomationML – die notwendige Flexibilität, die für Projekte im Industrie-4.0-Umfeld erforderlich ist. co

**Referenzen**

- [1] AutomationML part 1: IEC 62714–1 „Engineering data exchange format for use in industrial automation systems engineering – Part 1: Architecture and General Requirements“, 2014.
- [2] AutomationML part 2: IEC 62714–2 „Engineering data exchange format for use in industrial automation systems engineering – Automation markup language – Part 2: Role class libraries“, 2015.
- [3] Barth M., Biffi S., Drath R., Fay A., Winkler D.: „Bewertung der Offenheit von Engineering-Tools“, Journal, OpenAutomation, 3/2013, VDE Verlag, 2013.
- [4] Bauernhansl T., Tompel M., Vogel-Heuser B.: „Industrie 4.0 in Produktion, Automatisierung und Logistik“, Springer Vieweg, ISBN 978-3658046811, 2014.
- [5] Biffi S., Mätzler E., Wimmer M., Lüder A., Schmidt N.: „Linking and Versioning Support for AutomationML: A Model-Driven Engineering Perspective“, In: Proceedings of the 14th IEEE International Conference on Industrial Informatics (INDIN), 2015 (to appear).
- [6] Draht R. (ed): „Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PCLopen XML und Collada“, ISBN: 978-3-642-04673-5, VDI-Buch, Springer, 2010.
- [7] Draht R., Fay A., Barth M.: Interoperabilität von Engineering-Werkzeugen: Konzept und Empfehlungen für den Datenaustausch zwischen Engineering-Werkzeugen, In: Automatisierungstechnik 59(7), 2011.
- [8] Institute of Ergonomics, Manufacturing Systems and Automation at Otto-v.-Guericke University Magdeburg: Equipment Center for Distributed Systems, [http://www.iaf-bg.ovgu.de/technische\\_ausstattung\\_cvs.html](http://www.iaf-bg.ovgu.de/technische_ausstattung_cvs.html), last access July 2015.
- [9] Winkler D., Biffi S.: „Improving Quality Assurance in Automation Systems Development Projects“, In „Quality Assurance and Management“, Book Chapter, Intec Publishing, ISBN 979-953-307-494-7, 2012.

**Die Autoren:**

*Dietmar Winkler und Stefan Biffi (beide TU Wien, CDL-Flex) sowie Heinrich Steininger (logi.cals)*



**Kontakt**

Technische Universität Wien  
 Institut für Softwaretechnik und Interaktive Systeme  
 Christian Doppler Labor für ‚Software Engineering  
 Integration for Flexible Automation Systems‘ (CDL-Flex)  
 Wien  
 dietmar.winkler@tuwien.ac.at  
 stefan.biffi@tuwien.ac.at  
<http://cdl.ifs.tuwien.ac.at>

logi.cals GmbH  
 St.Pölten  
 heinrich.steininger@logicals.com  
<http://www.logicals.com>