Halle 7
Stand 174

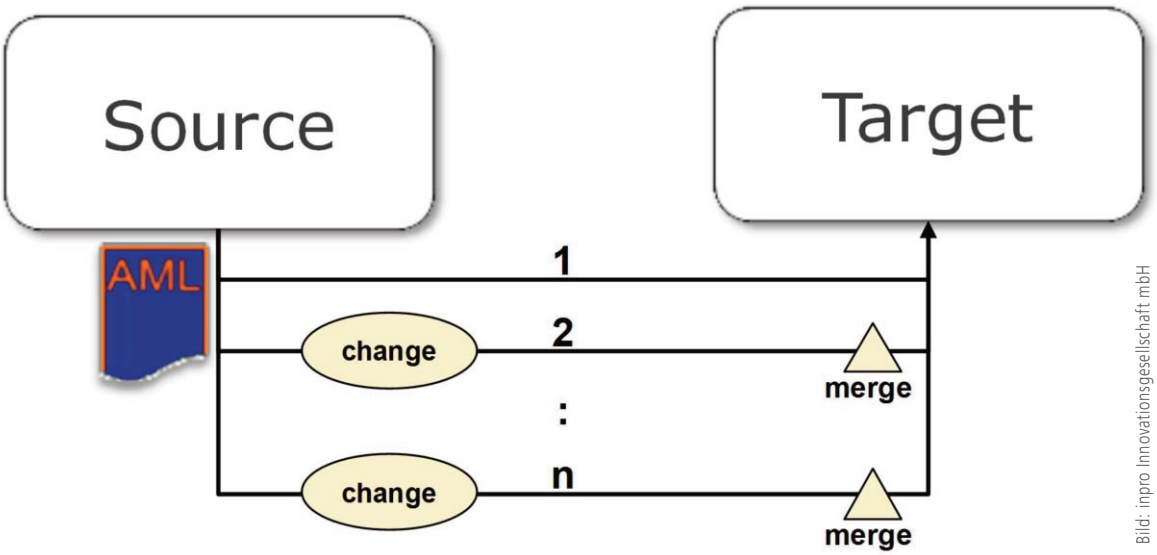

Bild 1: Iteratives Engineering

Bild: inpro Innovationsgesellschaft mbH

AutomationML – Engineering Workflow

Serie AutomationML Teil 11: Unterstützung von Engineering Workflows

Der Datenaustausch zwischen zwei Entwurfswerkzeugen während eines Entwicklungsprojektes ist oft keine einmalige Angelegenheit, sondern findet wiederholt statt. Objekte werden auch nach einem Datenaustausch beim Datenversender und beim Datenempfänger geändert, spezifiziert oder möglicherweise neu definiert. Bei einem wiederholten Datenaustausch in einem Entwicklungsprojekt sollten möglichst alle Änderungen als solche kenntlich gemacht werden, sodass das Zielwerkzeug einen selektiven Import der Daten durchführen kann und Änderungen mit dem eigenen Datenbestand vergleichen und in diesen sinnvoll integrieren kann. Ein Änderungs- und Versionsmanagement ist bei lose gekoppelten Werkzeugen, die lediglich über eine Datei-Schnittstelle und nicht über eine gemeinsame Datenbank verknüpft sind, in der Regel nicht möglich. Dieser Beitrag zeigt, wie auch für die dateibasierte Kopplung – mit Hilfe des AutomationML Austauschformates – ein Änderungs- und Versionsmanagement implementiert und damit komplexere Workflows unterstützt werden können, die über den einmaligen Datenaustausch weit hinausgehen.

Beim iterativen Engineering (Bild 1) werden Daten zwischen zwei Engineering-Werkzeugen während eines Entwicklungsprojektes wiederholt ausgetauscht. Der Datenaustausch ist unidirektional vom Quell- zum Zielsystem organisiert. Bei einem wiederholten Datenexport werden dabei vom Engineeringsystem, das die Datenquelle repräsentiert, zu jedem geänderten AutomationML-Objekt die entsprechenden Änderungsinformationen (mit) in die AutomationML-Datei geschrieben. Diese Informationen werden auf der Empfängerseite interpretiert, sodass eine Selektion und Integration der Daten in das Zielsystem erfolgen kann. Zur Kennzeichnung von Änderungen der Daten innerhalb eines Projektes bietet AutomationML folgende Konzepte an:

- ChangeMode – Attribute für jedes CAEX-Objekt
- Revision/Version – Information für jedes CAEX-Objekt
- Globally Unique Identifier (GUID) für alle AutomationML-Objekte
- Projekt Identifikation im 'Writer-Header'

Das 'ChangeMode'-Attribut erlaubt dem Versender, ein Objekt als neu ('create'), geändert ('change'), unverändert ('state') oder gelöscht ('delete') zu kennzeichnen. Soll das ursprüngliche Objekt auch bei Änderungen noch in der AutomationML Datei erhalten bleiben, besteht die Möglichkeit, das geänderte Objekt neu anzulegen und bei diesem und bei dem ursprünglichen Originalobjekt entsprechende Versionsinformationen zu hinterlegen. Der Versender der Daten muss bei einem wiederholten Datenexport auf

Auszug aus dem SPS-MAGAZIN, Fachzeitschrift für Automatisierungstechnik, Ausgabe 11/2013.

die ursprünglich erzeugten AutomationML-Dokumente (bzw. auf die zuletzt versendete Datei) zugreifen können, um Änderungsinformationen generieren zu können. Eine einfache Möglichkeit dies zu realisieren, ist die Verwendung eines projektspezifischen Dateiablageorts für die Versionskontrolle der versendeten AutomationML-Dateien durch den Versender. In diesem Verzeichnis können auch die nötigen Zuordnungsdateien (Endung .iaml in Bild 2) abgelegt werden, über die die Zuordnung der AutomationML-Objekte zu den ursprünglichen Systemobjekten erfolgen kann (Bild 2). Eine Voraussetzung zur effektiven Nutzung der beschriebenen Konzepte besteht darin, dass jedes AutomationML-Objekt eine eindeutige globale ID (GUID) erhält. Diese ID bleibt über den gesamten Lebenszyklus des Objektes erhalten. So behält ein AutomationML-Objekt, das bei einem wiederholten Datenaustausch als geändertes Objekt gekennzeichnet wird, die ursprüngliche GUID, die es bei seiner 'Geburt' erhalten hat. In einer Zuordnungstabelle kann diese GUID mit einem systemeigenen Identifizierungsschlüssel assoziiert werden, sodass ein Exporter die Änderungsinformationen für jedes Objekt anhand der Zuordnungstabelle und einem Vergleich mit der zuletzt erzeugten AutomationML-Datei generieren kann. Das Empfängerwerkzeug der Daten kann erkennen, ob ein AutomationML-Objekt zum ersten Mal gesendet wird, da es in diesem Fall den ChangeMode 'create' besitzt. In allen nachfolgenden

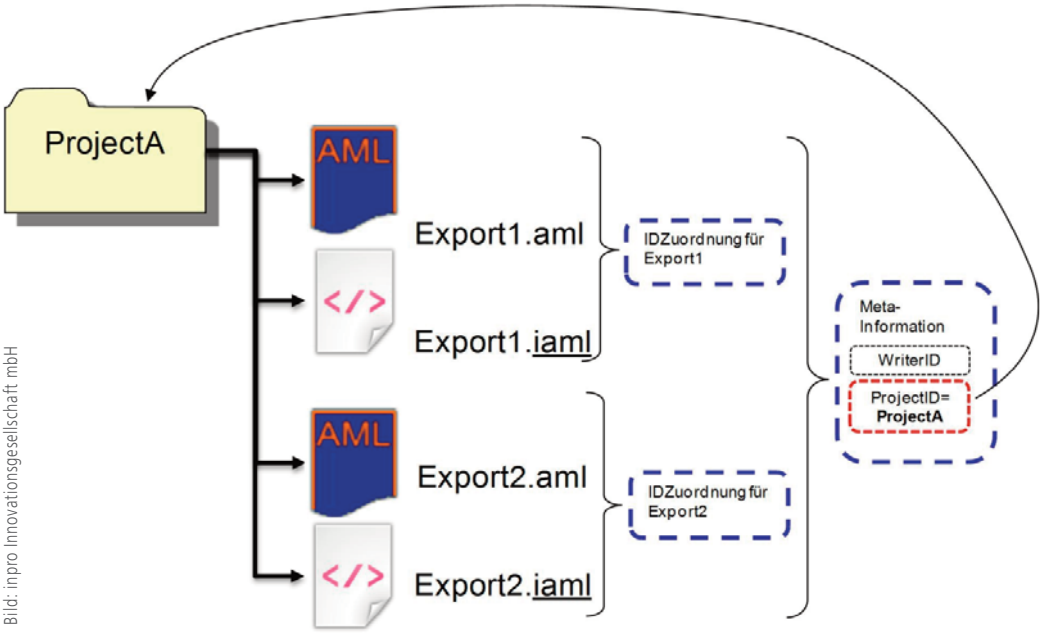

Bild: inpro Innovationsgesellschaft mbH

Bild 2: Organisation der Dateiverwaltung

Sendungen besitzt dasselbe Objekt einen anderen der möglichen ChangeMode-Werte ('change', 'state', 'delete'). Bei einem selektiven Import müssen Objekte mit einem ChangeMode-Wert von 'change' oder 'delete' besonders behandelt werden, da in der Regel die eingelesenen Daten seit dem letzten Import im Zielsystem weiter bearbeitet wurden. Bei geänderten Objekten (ChangeMode='change') muss darauf geachtet werden, dass keine Änderungen überschrieben werden und dadurch Planungsergebnisse verloren gehen. Auch bei gelöschten Objekten (ChangeMode='delete') ist Vorsicht geboten, wenn diese inzwischen mit Planungsdaten verknüpft wurden, die nicht automatisch mit gelöscht werden dürfen. Der Empfänger der Daten muss für die Wiedererkennung von importierten Objekten aus vorherigen Iterationen auf jeden Fall die ID-Zuordnungstabelle aufheben, welche alle Zuordnungen zwischen AutomationML-Objekt IDs (GUID) und System-Objekt IDs enthält. Es kann vorteilhaft sein, für den Vergleich von Änderungen das schon existierende Systemobjekt zunächst in ein AutomationML-Objekt zu transformieren, um den Vergleich auf Basis eines einheitlichen Formats

durchführen zu können. Bild 3 zeigt einen Lebenszyklus eines AutomationML-Objektes über mehrere Iterationsstufen. Im Beitrag zum Thema Datenkonsistenz [1] skizzieren die Autoren ein Systemkonzept für ein systemunabhängiges Werkzeug, das auf Grundlage der AutomationML-Dokumente und der enthaltenen Absender- und Änderungsinformationen die Datenkonsistenz in solchen Engineering Workflows überwachen und unterstützen kann.

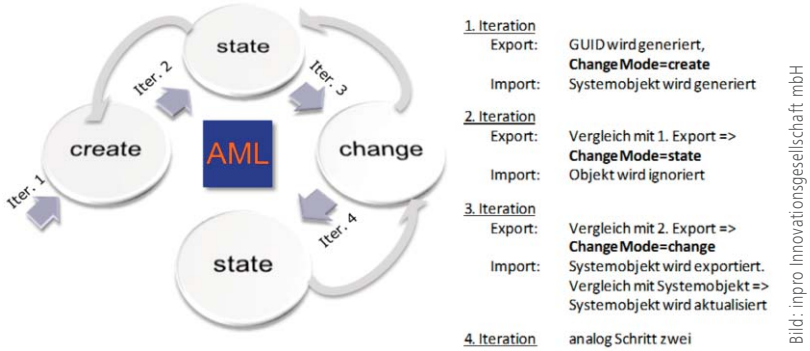

Bild: inpro Innovationsgesellschaft mbH

Bild 3: Lebenszyklus eines AutomationML-Objektes im iterativen Engineering

Roundtrip Engineering

Der Workflow des Roundtrip Engineering stellt eine Erweiterung des Iterativen-Engineerings dar und erlaubt das Ändern und Zurücksenden von importierten Daten an das Ursprungswerkzeug. Im Engineering tritt der Fall häufig auf, beispielsweise wenn ein generisches Objekt oder generische Merkmale eines Objektes spezifiziert werden, um den spezifizierten Wert danach in allen weiteren Planungsschritten benutzen zu können. Ein Beispiel hierfür wäre die Kalkulation eines Zeit- oder Kostendatums durch ein spezifisches Kalkulationstool. Auch in diesem Fall kann der Datenaustausch zwischen den Werkzeugen wiederholt stattfinden (Bild 4). In diesem Fall besitzt jedes der beteiligten Systeme sowohl Export- als auch Import-Funktionen und eine Möglichkeit zur Verwaltung der AutomationML-Dokumente und Zuordnungstabellen. Der Datenexport, den das Zielsystem durchführt, ist aber restriktiver als der Datenexport des Quellsystems. Das Zielsystem generiert beim Export AutomationML-Objekte, die an vorher eingelesene AutomationML-Objekte angefügt werden. Das generierte AutomationML-Dokument enthält daher immer AutomationML-Objekte, die auch in der zuerst importierten AutomationML-Datei enthalten waren. Idealerweise wird dafür dasselbe AutomationML-Dokument, das vorher empfangen wurde, genutzt. Nur so kann der ursprüngliche Sender erkennen, welche Objekte in der zurückkommenden Datei geändert/erweitert wurden. Zur Gewährleistung der Datenkonsistenz kann es besser sein, die Originalobjekte nicht direkt zu verändern, sondern Kopien zu erzeugen, zu verändern und als Revisionen bzw. Versionen des Originalobjektes zu versenden. AutomationML enthält mehrere Konzepte für das Roundtrip Engineering. Der Datenerzeuger und Versender kann einem Empfänger Vorgaben für die Spezifikation mitgeben, indem er AutomationML-Objekte mit 'RoleRequirements' und 'Attribute-Constraints' assoziiert. Der Empfänger kann Spezifikationen durch 'Verfeinerungen' und 'Klassen-Instanz-Relationen' mit entsprechenden spezifischen Klassenbibliotheken definieren. Für die Änderung von Objekten durch den Empfänger sind das 'ChangeMode'-Attribut und das 'Versions- und Revisionskonzept' nutzbar. Das Integrieren der zurückkommenden Daten erfolgt in ähnlicher Weise, wie im unidirektionalen iterativen Engineering. Für die Integration wird die Änderungsinformation, die entweder direkt durch das Setzen des 'ChangeMode'-Attributes oder durch 'Versions-' und 'Revisionsobjekte' angegeben ist, ausgewertet, mit den vorhandenen Daten verglichen und in den vorhandenen Datenbestand eingefügt.

Bild 4: Roundtrip Engineer-ing

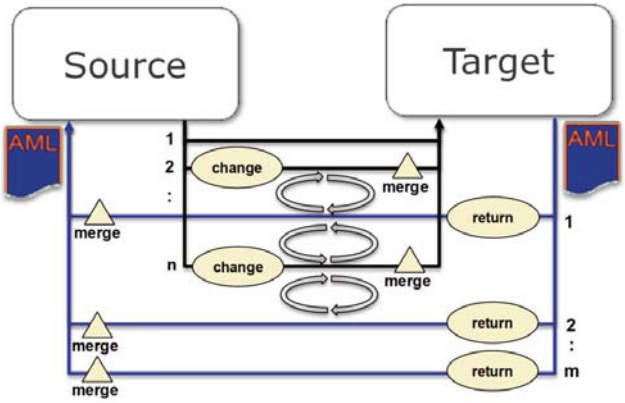

Bild: inpro Innovationsgesellschaft mbH

Fazit

Mit den ausdrucksstarken Beschreibungskonzepten von AutomationML ist es möglich, auch für lose gekoppelte Systeme ein Änderungs- und Versionsmanagement für definierte Engineering-Workflows zu implementieren. Die in diesem Beitrag dargestellten Konzepte und Workflows können auch für einen Verbund von mehreren Werkzeugen einzeln oder in Kombination verwendet werden. Weiterhin hat der Beitrag verdeutlicht, dass mit AutomationML-Dokumenten die Realisierung eines zentralen Datenpools für die redundanzfreie Verwaltung von Engineering-Daten möglich ist. Für diesen Fall müssen die beschriebenen Modellierungskonzepte des Formates zur Änderungs- und Versionskontrolle genutzt und darüber hinaus muss lediglich eine wechselseitige Zugriffskontrolle für die genutzten AutomationML-Dokumente implementiert werden. ■

Literatur

[1] R. Drath, B. Schröter, M. Hoernicke, Datenkonsistenz im Umfeld heterogener Engineering-Werkzeuge in VDI Automation 2011.

www.automationml.org

Autor: Josef Prinz, Senior-Experte Digitale Fabrik, inpro Innovationsgesellschaft mbH

Autor: Dr.-Ing. Lorenz Hundt, Projektingenieur Produktionssyst. und Informationsprozesse, inpro Innovationsgesellschaft mbH

Auszug aus dem SPS-MAGAZIN, Fachzeitschrift für Automatisierungstechnik, Ausgabe 11/2013.