

DIGITALE PRODUKTION VIA ENTERPRISE APPLICATION INTEGRATION

Dipl.-Inform. Tobias Meisen
Dipl.-Inform. Philipp Meisen
Dr.-Ing. Daniel Schilberg
Prof. Dr. rer. nat. Sabina Jeschke

1. Einleitung

Schon früh wurde im Unternehmensumfeld die Notwendigkeit erkannt unterschiedliche im Unternehmen verwendete Applikationen zu verknüpfen. Die zu diesem Zweck entwickelten Konzepte wurden unter dem Sammelbegriff Enterprise Application Integration (EAI) bekannt. EAI hat zum Ziel, die IT gestützten Geschäftsfunktionen entlang der Wertschöpfungskette eines Unternehmens unabhängig von Plattform, Architektur und Generation der verwendeten Applikationen zu integrieren [6].

Unternehmen aus dem produktionstechnischen Umfeld nutzen bereits seit einigen Jahren Verfahren, die aus dem Umfeld der EAI stammen. So wird die unternehmensweite Ressourcenplanung in den meisten Unternehmen mittels ERP Systemen [7], die EAI Konzepte zur Daten- und Applikationsintegration nutzen, realisiert. Auch der Einsatz von Business Intelligence (BI) Systemen zur prozessübergreifenden und wertschöpfungsorientierten Informationsgewinnung in Produktionsbetrieben [4] wird erst durch den Einsatz von Integrationsverfahren ermöglicht. Die einheitliche Betrachtung anderer Unternehmensprozesse als die der Geschäftsfunktionen, beispielsweise von Simulationsprozessen in der Forschung und Entwicklung, findet allerdings selten statt [10]. Dabei sind Fertigungsverfahren wie Schweiß-, Umform- oder Wärmebehandlungsprozesse, die bei der Produktion von Gütern durchlaufen werden, größtenteils durch Simulationen beschreibbar. Diese sind jedoch auf einzelne Prozesse spezialisiert und verfügen weder über standardisierte Schnittstellen noch standardisierte Datenformate. Es handelt sich um sogenannte Insellösungen. Zur durchgängigen Kopplung zu einem Simulationsprozess müssen die einzelnen Simulationsergebnisse daher derzeit manuell überprüft und für nachfolgende Simulationen angepasst werden. Dies erfordert einen hohen Zeitaufwand und birgt ein enormes Fehlerpotential. Außerdem ist die Nutzung gängiger EAI Lösungen für solche Integrationsvorhaben wegen unzureichender Unterstützung nicht möglich. So werden beispielsweise keine großen Datenmengen unterstützt, die aber in Simulationsprozessen charakteristisch sind. Weiterhin fehlen Standards, die Integrationsbemühungen im Simulationsumfeld vorantreiben.

In diesem Beitrag wird ein Framework beschrieben, das die Simulation eines Fertigungsprozesses unter Verwendung bestehender Insellösungen ermöglicht. Im Fokus steht dabei nicht die technische Kopplung der Applikationen, die mittels moderner Middleware Techniken

[8][9] gelöst werden kann, sondern die Integration der jeweiligen durch die Applikation generierten Daten. Das Framework wird im Rahmen des Exzellenzclusters „Integrative Produktionstechnik für Hochlohnländern“ im Teilprojekt „Integrierte Plattform für verteilte numerische Simulation“ entwickelt. Die Validierung des Frameworks erfolgt anhand der Simulation des Fertigungsprozesses einer Linepipe, die im Rahmen eines weiteren Teilprojektes des Exzellenzclusters durchgeführt wird.

In Kapitel 2 wird zunächst der Stand der Technik dargestellt, um in Kapitel 3 den Anwendungsfall für das in diesem Beitrag vorgestellte Framework zu beschreiben. In Kapitel 4 folgt die Beschreibung der Frameworkarchitektur, in Kapitel 5 die Vorstellung der für den Anwendungsfall notwendigen Erweiterungen am Framework. Kapitel 6 schließt diesen Beitrag mit einem Fazit ab.

2. Stand der Technik

Datenintegration und damit auch EAI ist bereits seit den Achtzigern, spätestens aber seit den neunziger Jahren eines der wichtigsten Themen, wenn es um die Beantwortung applikationsübergreifender Fragestellungen geht [26]. Grundsätzlich lassen sich heute eine Vielzahl verschiedener Datenintegrationsprodukte finden, welche in unterschiedlichen Anwendungsgebieten nutzen finden. Im Allgemeinen lassen sich diese in drei Hauptbereiche [5] unterteilen (vgl. Abbildung 1):

- Datenverdichtung (Data Consolidation)
- Datenvereinigung (Data Federation)
- Datenverbreitung (Data Propagation)

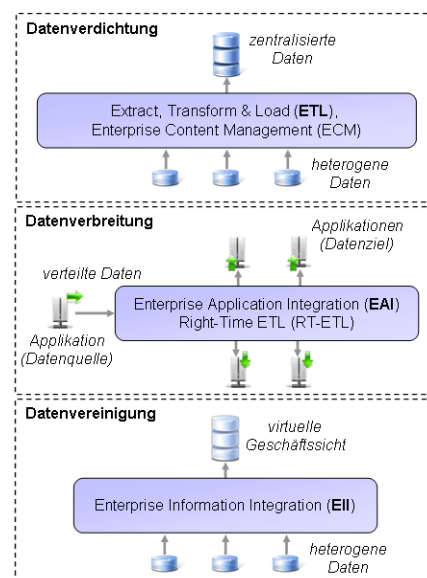


Abbildung 1: Hauptbereiche der Datenintegration [5]

Im operationalen Bereich wird der Bereich der Datenverbreitung zur applikationsübergreifenden Nutzung von Daten angewendet. Diese wird häufig durch EAI realisiert. Wie in [5] dargestellt, fokussiert sich EAI im Wesentlichen auf Nachrichten und Business Transaktionen, also kleinere Datenmengen, die zwischen verschiedenen Applikation ausgetauscht werden. Ein modernes Architekturkonzept, dass im Rahmen dieses Beitrags aufgegriffen wird und zur Realisierung von EAI verwendet wird, ist der Enterprise Service Bus (ESB). Es wurde im Rahmen des Aufkommens von servicebasierten Ansätzen entwickelt [11]. Grundgedanke von ESB ist die Bereitstellung von Diensten innerhalb eines Systems, ähnlich der Verwendung von Integration Brookern [12]. Abbildung 2 zeigt den schematischen Aufbau eines ESB.

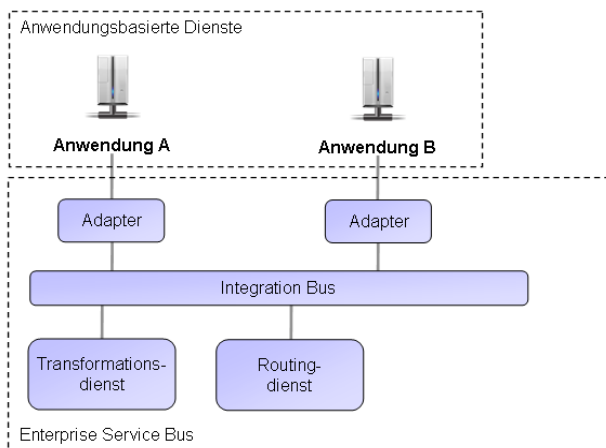


Abbildung 2: Schema eines Enterprise Service Bus

Ein Dienst stellt eine fachliche oder technische Funktionalität zur Verfügung, mit der die Geschäftsprozesse unterstützt werden. Alle Dienste sind über den Integration Bus miteinander verbunden. Transformationsdienste stellen allgemeine Funktionen zur Verfügung, um Daten eines Formats und Modells in ein anderes Format und Modell zu überführen. Routingdienste hingegen dienen zur Übertragung von Daten an andere Dienste. Adapter letztendlich nutzen Transformations- und Routingdienste, um über den Service Bus bereitgestellte Daten in das Format und Modell einer Anwendung zu überführen.

Transformationsdienste fördern somit die Wiederverwendung von implementierten Datentransformationen. Der Vorteil einer ESB Lösung liegt in der losen Kopplung der einzelnen Dienste. Nachteilig ist jedoch die fehlende physische Datenkopplung [13]: Sollen erfasste Daten nachträglich ausgewertet oder analysiert (z.B. mittels Datenexplorationstechniken wie OLAP oder Data Mining) werden, so müssen diese erneut von einer Datenquelle ausgelesen und transformiert werden, wodurch eine historische oder Langzeit Datenauswertung nicht möglich ist.

Zur Realisierung einer solchen einheitlichen Betrachtung über alle Daten hinweg müssen andere Bereiche der Datenintegration herangezogen werden (vgl. Abbildung 1). Eine Lösung für die Schaffung einer einheitlichen Sicht auf Daten ist die Datenvereinigung, die in dem Umfeld der Enterprise Information Integration (EII) untersucht wird. Hierbei ist es das Ziel, Daten aus unterschiedlichen Datenquellen unter einer uniformen Schnittstelle zu einen

[5][27]. Diese Schnittstelle kann dann genutzt werden, um Anfragen an diese Datenquellen zu formulieren und auszuwerten. Da die Anfrage auf den unterschiedlichen Datenquellen ausgeführt wird und keine Konsolidierung innerhalb einer Datenquelle erfolgt, handelt es sich bei diesem Ansatz um die Implementierung einer virtuellen Sicht auf die Daten. Hierzu ist es jedoch erforderlich, dass die Daten in verteilten Datenspeichern zur Verfügung stehen und auf diese zugegriffen werden kann. Trifft dies nicht zu müssen Techniken der Datenverdichtung angewendet werden. Unter Datenverdichtung wird das Zusammenführen unterschiedlicher Daten in eine gemeinsame, einheitliche Datenstruktur verstanden. Mittels Extract Transform Load (ETL) wird ein gängiger Prozess für die Integration von Daten beschrieben [28]. ETL umfasst die Extraktion der Daten aus einer oder mehreren, meistens operativen, Datenquelle, die Transformation des Datenformats und -modells in das Zielschema und das Laden in die Zieldatenbank.

3. Anwendungsfall

Bei der Fertigung einer Linepipe kommen unterschiedliche Fertigungsverfahren zum Einsatz. Im Rahmen des Anwendungsfalls werden die Fertigungsverfahren durch für diese Verfahren spezialisierte Werkzeuge simuliert. Den zugrundeliegenden Fertigungsprozess stellt Abbildung 3 dar.

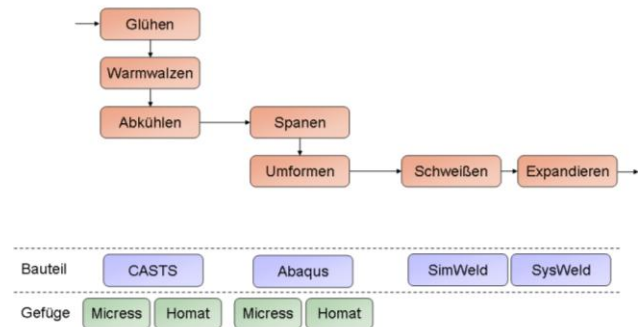


Abbildung 3: Fertigungsprozess Linepipe

Zunächst wird das Glühen, Warmwalzen und Abkühlen des Bauteils durch die von Access entwickelte Applikation CASTS simuliert. Anschließend wird das Spanen und Umformen mit Abaqus (SIMULIA) dargestellt. Das Schweißen und Expandieren der Linepipe wird in SimWeld, einer Entwicklung des ISF der RWTH Aachen und der Software SysWeld, eine Entwicklung der ESI-Group, simuliert [14]. Außerdem findet eine Simulation der Änderungen im Gefüge des Bauteils mit den Simulationen Micress [2] und Homat [3], beide Entwicklungen von Access, statt. Insgesamt umfasst der Anwendungsfall somit sechs unterschiedliche Simulationen, denen jeweils unterschiedliche Formate und Modelle zugrunde liegen.

Neben diesem Anwendungsfall werden im Rahmen des Projektes „Integrierte Plattform für verteilte numerische Simulation“ vier weitere, hier nicht näher erläuterte, Anwendungsfälle untersucht, welche die Basis für die Anforderungen an das in diesem Beitrag vorgestellte Framework bilden. Diese Anforderungen wurden bereits in [15] untersucht und beschrieben und in [16] vervollständigt. Zentrale Anforderung ist dabei die Möglichkeit zur

Datenverbreitung sowie die Notwendigkeit nach einer prozessorientierten Datenverdichtung (vgl. Abbildung 1) zur späteren Visualisierung und Analyse der im Prozess gesammelten Daten. Ein weiteres wichtiges Ziel ist es, neue Simulationsprozesse einfach abzubilden und neue Simulationswerkzeuge schnell einzubinden, ohne größere Anpassungen an der Anwendung vornehmen zu müssen.

4. Architektur des Frameworks

4.1 Systemarchitektur

Basierend auf den in Kapitel 3 geschilderten Anforderungen wurde die Architektur des Frameworks entwickelt. Sie ist angelehnt an das ESB Architekturkonzept, wobei die Möglichkeit zur Datenverdichtung durch Einführung eines zentralen Datenspeichers realisiert wurde [25]. Die Systemarchitektur ist in Abbildung 4 dargestellt.

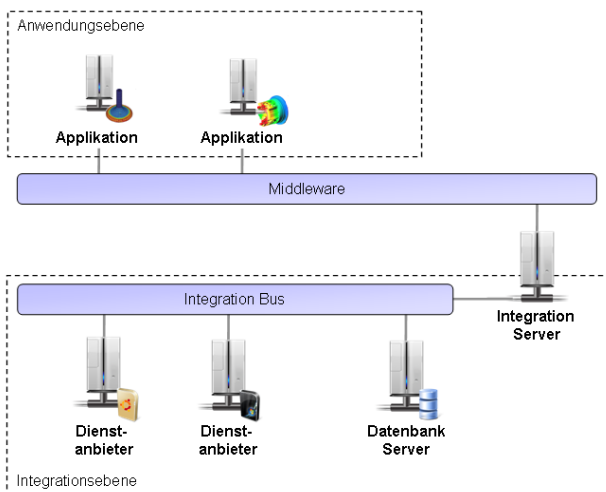


Abbildung 4: Systemarchitektur des Frameworks

Die Kommunikation mit den Applikationen erfolgt über eine Middleware. Im Rahmen des Anwendungsfalls wurde die anwendungsorientierte Middleware Condor [1][24] verwendet. Routingdienste, wie in einem ESB üblich, sind kein Bestandteil des Frameworks, da diese Funktionalität durch die Middleware gekapselt wird.

Das Framework wird zur Realisierung der Integrationssebene verwendet. Dienstanbieter stellen unterschiedliche Dienste zur Integration, Extraktion und Transformation von Daten bereit und sind direkt mit dem Integration Bus verbunden. Die Anbindung erfolgt über die Java RMI Technologie [17] und ist damit unabhängig vom verwendeten Betriebssystem. Einen zentralen Unterschied zum ESB Architekturkonzept stellt die Verwendung eines Integration Servers und eines Datenbank Servers dar. Der Integration Server empfängt alle eingehenden Daten von der Middleware, analysiert sie und stellt sie anschließend über den Integration Bus den einzelnen Dienst Anbietern zur Verfügung. Diese greifen die benötigten Daten ab und verarbeiten sie weiter. Nach jedem abgeschlossenen Verarbeitungsschritt prüft der Integration Server die Konsistenz der Daten und ermittelt, welcher Bearbeitungsschritt für die gegebenen Daten als nächstes durchgeführt werden muss. Der Integration Server ist somit die zentrale Kontrollinstanz des Datenintegrations- und Datenextraktionsprozesses.

Als zentraler Datenspeicher wird ein Datenbank Server verwendet, der ebenfalls mit dem Integration Bus verbunden ist. Sowohl die Dienstanbieter als auch der Integration Server greifen direkt auf den Datenbank Server zu. Hierdurch wird eine Datenverdichtung und somit eine nachgelagerte Analyse der über den Prozess gesammelten Daten ermöglicht.

4.2 Softwarearchitektur

Das Framework umfasst die drei Hauptkomponenten:

- Integration Server
- Service Provider
- Client

die im Folgenden im Detail beschrieben werden. Zur Kommunikation zwischen den Komponenten existiert außerdem eine Codebase Komponente, in der komponentenübergreifende Funktionalität gekapselt ist. Die Hauptkomponenten sind in Abbildung 5 dargestellt und werden im Folgenden erläutert.

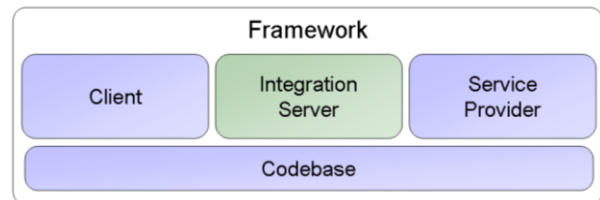


Abbildung 5: Hauptkomponenten des Frameworks

Client

Die Client Komponente dient zur Realisation der Kommunikation zwischen Middleware und Integration Server. Sie enthält Adapter, die eine Weiterleitung der Anfragen an den Integration Server über unterschiedliche Quellen, bspw. TCP/IP, RMI oder SOAP [18], ermöglichen. Für einen konkreten Anwendungsfall, der nicht durch die bereits integrierten Technologien abgedeckt ist, bietet die Client Komponente Grundgerüste an, die zur Implementierung eines eigenen Adapters verwendet werden können (vgl. Kapitel 5).

Integration Server

Die Integration Server Komponente enthält die Implementierung eines Integration Servers. Ein Integration Server fungiert als Dienstanbieter für die einzelnen Clients und stellt Integrations-, Extraktions- und Konvertierungsdienste bereit. Für jeden dieser Dienste ist ein Dienstprozess im Integration Server hinterlegt. Stellt ein Client eine Anfrage, so wird der zugrundeliegende Dienstprozess gestartet und abgearbeitet. Ein Dienstprozess beschreibt in Form eines gerichteten Ablaufes, welche Dienste abgearbeitet werden müssen, um die vom Client angeforderte Funktionalität bereitzustellen. Die im Framework realisierten Dienstprozesse sind in Abbildung 6 dargestellt.

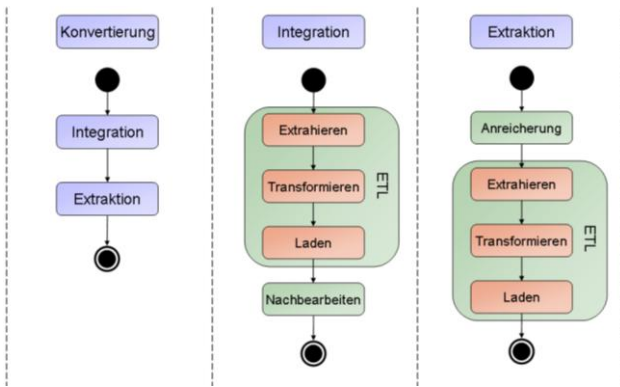


Abbildung 6: Dienstprozesse im Framework

Der Konvertierungsprozess wird durch den Integrations- und Extraktionsprozess, die beide auf einem erweiterten ETL Prozess basieren, definiert. Während dem Integrationsprozess eine Nachbearbeitung der integrierten Daten nachgelagert ist, wird bei dem Extraktionsprozess vor dem eigentlichen ETL Prozess eine Datenanreicherung durchgeführt. Die in Abbildung 6 grün markierten Ablaufschritte sind Funktionalitäten, die vom Dienstanbieter innerhalb der Integrationsebene bereitgestellt werden. Der Integration Server dient dabei als Vermittler [19] und vermittelt die entsprechenden Daten an denjenigen Dienstanbieter, der die geforderte Funktionalität und Kapazität bereitstellt. Somit sind die für die Datenintegration relevanten Algorithmen, die abhängig vom Anwendungsfall sind, in den konkreten Dienstanbietern gekapselt.

Außerdem realisiert der Integration Server eine prozessbezogene Datenintegration. Der Integration Server überwacht die Zuordnung der Daten zu dem Prozessschritt und vermittelt den Prozesskontext an die Dienstanbieter. Das zugrundeliegende Datenschema ist in Abbildung 7 dargestellt.

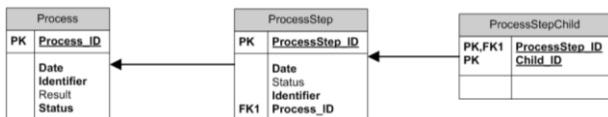


Abbildung 7: Datenschema zur prozessbezogenen Datenintegration

Ein Prozess wird darin durch einen Namen (Attribut Identifier) gekennzeichnet. Ebenso verfügt jeder Prozessschritt über einen frei wählbaren Namen (Attribut Identifier). Der Prozessablauf wird abgebildet, indem die Nachfolger eines Prozessschrittes gespeichert werden. Dabei kann ein Prozessschritt beliebig viele Nachfolger aufweisen, um komplexe Prozesse, wie beispielsweise die Bereitstellung von Simulationsdaten auf Makro- und Mikroebene, zu berücksichtigen.

Service Provider

Die Service Provider Komponente stellt die grundlegende Funktionalität für Dienstanbieter im Framework bereit. Die Implementierung eines konkreten Dienstanbieters hängt dabei vom Anwendungsfall ab. So basiert beispielsweise die Integration von FEM-Daten auf einem anderen Datenschema als die von Molekülstruktur-Daten, obwohl es sich um denselben physikalischen Gegenstand und vergleichbare physikalische Größen handelt.

Das Framework bietet Schnittstellen zu gängigen ETL-Werkzeugen, wie beispielsweise dem Pentaho Data Integrator (PDI) [21]. Somit kann die Integration und Extraktion von Daten auf Basis dieser, im Umfeld von ETL bereits etablierten, Werkzeugen erstellt werden. Umgekehrt ist es jedoch auch möglich, andere Werkzeuge oder Frameworks zur Integration und Extraktion zu nutzen, falls dies im konkreten Anwendungsfall (vgl. Kapitel 5) sinnvoll und nötig ist.

Wie in Abbildung 6 dargestellt unterstützt das Framework neben Diensten zur Bereitstellung eines ETL-Prozesses auch Dienste zur Nachbearbeitung und Anreicherung von Daten. Der Nachbearbeitungsdienst ermöglicht dabei beispielsweise die Implementierung von Plausibilitätskriterien, die von den integrierten Daten unabhängig von ihrer Ursprungsquelle erfüllt werden müssen. Diese können in einem Nachbearbeitungsdienst implementiert werden. Innerhalb der Anreicherung werden Datentransformationen durchgeführt, um die im zentralen Datenspeicher hinterlegten Daten so aufzubereiten, dass die an sie gestellten Bedingungen zur Extraktion erfüllt sind. Innerhalb des Frameworks ist eine adaptive, auf Ontologien basierende, Datenanreicherung umgesetzt [22]. Hierdurch kann basierend auf den gegebenen (Ist-Zustand) und den benötigten Daten (Soll-Zustand) eine Transformationskette bestimmt werden, die den Ist-Zustand in den Soll-Zustand überführt.

5. Anwendung des Frameworks

Im Rahmen des in Kapitel 3 beschriebenen Anwendungsfalles und den Anforderungen aus vier weiteren betrachteten Anwendungsfällen wurde eine auf dem Framework basierende Applikation implementiert. Dazu wurde zunächst ein Datenschema zur Speicherung der von den Simulationen gelieferten Daten definiert. Die im Anwendungsfall betrachteten Simulationen basieren auf der Finite-Elemente-Methode [20], so dass das Datenschema für diese Art von Daten ausgelegt wurde. Abbildung 8 zeigt einen Auszug aus dem zugrundeliegenden Datenschema in UML Notation. Dieses Datenschema wurde als Objektmodell in der Anwendung, wie auch in der relationalen Datenbank abgebildet.

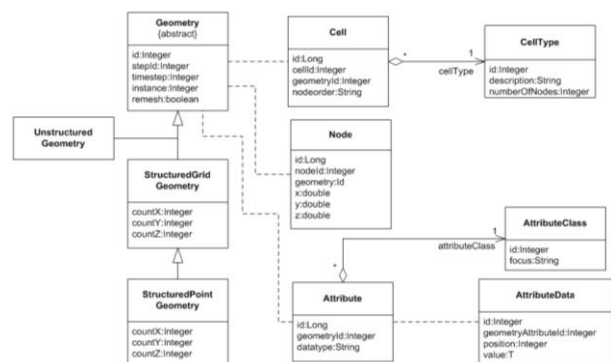


Abbildung 8: Datenschema für FEM-Simulationen

Die zentrale Entität in diesen Simulationen ist die Geometrie des zu simulierenden Bauteils, die aus Knoten, Zellen und Attributen besteht. Attribute verfügen über Attributwerte, die abhängig von der Attributklasse der gesamten Geometrie einzelnen Zellen oder Knoten zugeordnet sind.

Die im Rahmen des Anwendungsfalls spezifizierten Integrationsdienste lesen die von der Simulation gegebenen Geometriedaten ein, transformieren sie in das zentrale Datenschema und laden das Ergebnis in die Datenbank. Die Extraktionsdienste hingegen reichern die Daten zunächst derart an, dass die Daten in einer für das Zielformat gültigen Form vollständig vorliegen. Anschließend lesen sie die Geometriedaten aus der zentralen Datenbank aus, transformieren sie in das geforderte Format und laden die Daten in die Zieldatei oder -datenbank. Dieses Vorgehen soll exemplarisch am Beispiel der Konvertierung von Simulationsergebnissen der Simulation CASTS in eine Eingabedatei der Simulation Abaqus erläutert werden. Die Simulation CASTS speichert ihre Simulationsergebnisse im Format des Visualization Toolkits (VTK) [23]. Daher wurde ein Integrationsdienst für VTK, basierend auf der vom VTK Entwickler bereitgestellten Programmierschnittstelle, in der Anwendung implementiert. Die Bereitstellung der Dateien und der Datenbankzugriff sind bereits im Framework realisiert. Außerdem wurde ein Extraktionsdienst für das Abaqus Eingabeformat entwickelt, wobei hier aufgrund einer fehlenden Programmierschnittstelle auf das erwähnte ETL Werkzeug PDI zurückgegriffen wurde. Der implementierte PDI Job ist in Abbildung 9 dargestellt.

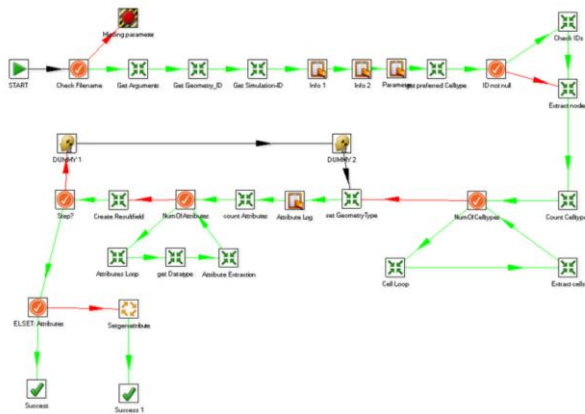


Abbildung 9: Abaqus Extraktion als PDI Transformation

Zur Realisierung der adaptiven Datenanreicherung wurden verschiedene Datentransformationen für FEM-Daten in die Anwendung implementiert. Beispiele für solche Transformationen sind die Umwandlung von Attributeinheiten, die Ableitung von Attributen aus bestehenden Attributen, das Verschieben der Geometrie im Raum, die Veränderung von Zelltypen einer Geometrie (bspw. Hexaeder zu Tetraeder) oder die Neu Nummerierung von Knoten und Zellen. In Abhängigkeit von den vorliegenden Daten kann die Anwendung die vorhandenen Daten mittels der adaptiven Datenanreicherung so transformieren, dass eine Datenextraktion ermöglicht wird. Im Falle des oben genannten Beispiels wird so eine Neu Nummerierung der Knoten und Zellen veranlasst. Dies ist notwendig, da die Zell- und Knotennummerierung in VTK bei 0, in Abaqus hingegen bei 1 beginnt. Ebenso werden in CASTS Vektoren in ihre einzelnen Komponenten zerlegt und als Attributwerte von Knoten abgespeichert, wohingegen Abaqus die Angabe der vollständigen Vektoren erwartet. Die Anwendung ist aufgrund der Datenanreicherung in der Lage, diese Anforderung selbstständig

zu erkennen und dementsprechend die Daten als vollständigen Vektoren und nicht als Zerlegung in seine einzelnen Komponenten bereitzustellen. Die Konvertierung von CASTS Daten in Abaqus Daten für den Anwendungsfall ist in Abbildung 10 vereinfacht dargestellt.

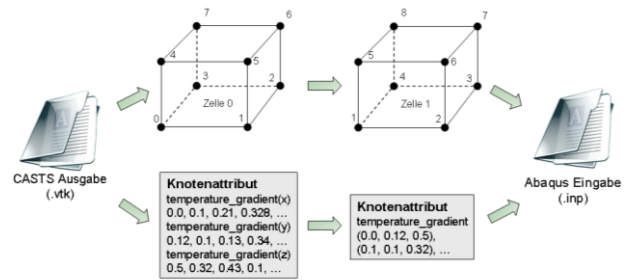


Abbildung 10: Beispiel einer Datenanreicherung

6. Fazit

Die Entwicklung des in diesem Beitrag vorgestellten Frameworks stellt einen wichtigen Schritt in der Etablierung der virtuellen Produktion dar, da es ermöglicht wird, Fertigungsprozesse Schritt für Schritt unter Verwendung spezialisierter Werkzeuge zu simulieren. Diese ganzheitliche auf spezialisierten Werkzeugen basierende Simulation findet dabei ohne Datenverluste oder zeitintensive manuelle Datenübernahmen von einem Werkzeug ins nächste statt. Ermöglicht wird dies durch das vorgestellte Framework, welches einen Weg eröffnet, die bisher unabhängig voneinander entwickelten, für einzelne Fertigungsverfahren oder Methoden spezialisierten Simulationen miteinander zu koppeln und die während der Simulationen generierten Daten einheitlich und prozessorientiert zu integrieren. Neben der Integration weiterer Simulationen ist es in einem nächsten Schritt erforderlich, die Domäne der betrachteten Simulationen um Maschinen- und Fabriksimulationen zu erweitern, um eine ganzheitliche Simulation von Fertigungsprozessen zu ermöglichen. Die größte Herausforderung liegt dabei in der Schaffung eines zentralen Datenschemas, welches es ermöglicht, die Daten einheitlich und unter Berücksichtigung ihrer Bedeutung im Gesamtkontext der Fabrik-, Maschinen und Materialebene abzubilden. Aufgrund der vorgestellten Methodik ist es dabei nicht erforderlich die Anwendungen auf dieses Datenschema anzupassen. Dies realisiert die Integrationsanwendung, die, basierend auf dem Framework, zu entwickeln ist. Durch die einheitliche Datensicht und die detaillierte Datenaufzeichnung auf Prozessebene ermöglicht das Framework in Zukunft, die Ergebnisse unterschiedlicher Simulationsprozesse und Simulationen miteinander zu vergleichen. Weiterhin können nun Rückschlüsse auf Fehlerquellen gezogen werden, was in der Vergangenheit nur unter hohen Zeit- und Kostenaufwand realisierbar war. Um dies zu realisieren, werden als nächstes die interessanten Merkmale (Performance Indicators) identifiziert und anschließend innerhalb der Anwendung bereitgestellt werden. Die Herausforderungen liegen hier vor allem bei der Entwicklung der benötigten Datenexplorationsverfahren sowie den Visualisierungstechniken.

7. Danksagung

Die vorgestellten Arbeiten wurden von der Deutschen Forschungsgemeinschaft DFG im Rahmen des Exzellenzclusters "Integrative Produktionstechnik für Hochlohnländer" gefördert.

8. Literatur

- [1] Thain, Douglas, Tannenbaum, Todd, Livny, Miron, 2005, „Distributed Computing in Practice: The Condor Experience“, *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, pages 323-356.
- [2] Laschet, G., Neises, J. and Steinbach, I., 1998, „Micro- Macrosimulation of casting processes“, 4ième école d'été de "Modélisation numérique en thermique", C8 1-42, Porquerolles.
- [3] Laschet, G., 2002, „Homogenization of the thermal properties of transpiration cooled multi-layer plates“, *Computer Methods in Applied Mechanics and Engineering*, Volume 191, Issues 41-42, Pages 4535-4554.
- [4] Panian, Z., 2005, „Supply chain intelligence in e-business environment“, *Proceedings of the 9th WSEAS International Conference on Computers*, Article No.: 85 (ISBN: 960-8457-29-7), Athens, Greece.
- [5] White, C., 2005, „Data Integration: Using ETL, EAI, and EII Tools to Create an Integrated Enterprise“, *TDWI Report Series*.
- [6] Conrad, C., Hasselbring, W., Koschel, A. and Tritsch, R., 2005, „Enterprise Application Integration: Grundlagen - Konzepte - Entwurfsmuster – Praxisbeispiele“, *Spektrum Akademischer Verlag*, 1. Auflage.
- [7] Gronau, N., 2010, „Enterprise Resource Planning“, Oldenbourg; 2. Auflage.
- [8] Myerson, J. M., 2002, „The Complete Book of Middleware“, *Crc Pr Inc*, 1. Auflage.
- [9] Serain, D., 2002, „Middleware and Enterprise Application Integration: The Architecture of E-Business Solutions“, *Springer*, 2. Auflage.
- [10] Schmitz, G.J., Prah, U., 2009, „Toward a Virtual Platform for Materials Processing“, *JOM Vol. 61 No.5* p. 19-23.
- [11] Chappell, D. A., 2004, „Enterprise Service Bus: Theory in Practice“, *O'Reilly Media*.
- [12] Schulte, Roy W., 2002, „Predicts 2003: Enterprise Service Buses Emerge“, *Gartner*.
- [13] Rademakers, Tijs, Dirksen, Jos, 2008, „Open-Source Esbs in Action“, *Manning Publications Co.*
- [14] Rossiter, E., Mokrov, O., Dilthey, U., 2007, „Integration des Simulationspaketes SimWeld in FEM-Analyseprogramme zur Modellierung von Schweißprozessen“, *Sysweld Forum*, Weimar, Deutschland.
- [15] Schilberg, D., Meisen, T., Henning, K., 2009, „Ontology Based Semantic Interconnection of Distributed Numerical Simulations for Virtual Production“, *IEEE 16th International Conference on Industrial Engineering and Engineering Management (IE&EM)*, Beijing (PR China).
- [16] Schilberg, D., 2010, „Architektur eines Datenintegrators zur durchgängigen Kopplung von verteilten numerischen Simulationen“, *Fortschritt-Berichte VDI Reihe 10 Informatik/Kommunikation, Band 807*, Düsseldorf: VDI Verlag.
- [17] Grosso, W., *Java RMI (Java Series)*, 2001, *O'Reilly Media*, 1. Auflage.
- [18] Kashyap, V., Bussler, C., Moran, M., 2008, „The Semantic Web: Semantics for Data and Services on the Web (Data-Centric Systems and Applications)“, *Springer, Deutschland*, 1. Auflage.
- [19] Gamma, E., Helm, R., Johnson, R. E., 1995, „Design Patterns. Elements of Reusable Object-Oriented Software.“, *Addison-Wesley Longman, Amsterdam*.
- [20] Steinke, Peter, 2010, „Finite-Elemente-Methode: Rechnergestützte Einführung“, *Springer*, 3. Auflage.
- [21] ETL White Paper - Advanced ETL with Pentaho Data Integration, 2006, *Breadboard BI*.
- [22] Meisen, T., Schilberg, D., Henning, K., 2009, „Planner Based Data Integration for Simulation Chains in Virtual Production“, *Proceedings International Conference on Science, Technology and Innovation for Sustainable Well-Being (STISWB)*, Hrsg. v. Mahasarakham University, Thailand: Khon Kaen, Thailand: Klung Na Na Vithya Press Limited Partnership.
- [23] Schroeder, W., Martin, K., Lorensen, B., 2004, „Visualization Toolkit“, *Kitware Inc*; 3. Auflage.
- [24] Cerfontaine, P., Beer, T., Kuhlen, T., Bischof, 2008, „Towards a Flexible and Distributed Simulation Platform“, *Lecture Notes in Computer Science (Springer), Computational Science and Its Applications - ICCSA 2008, Proceedings Part 1*, p. 867 – 882, *Springer*.
- [25] Schilberg, Daniel, Gramatke, Arno, Henning, Klaus, 2008, „Semantic Interconnection of Distributed Numerical Simulations Via SOA“, In: *Proceedings World Congress on Engineering and Computer Science 2008*, Hrsg. v. International Association of Engineers: Hong Kong: Newswood Limited, S. 894-897.
- [26] Halevy, Alon, Anand, Rajaraman, Ordille, Joann, 2006, „Data integration: the teenage years“, *VLDB'2006: Proceedings of the 32nd international conference on Very large data bases In VLDB'2006: Proceedings of the 32nd international conference on Very large data bases*, pp. 9-16.
- [27] Bernstein, Philip A., Haas, Laura M., 2008, „Information integration in the enterprise“, *Communications of the ACM archive Volume 51, Issue 9 (September 2008)*, pp. 72-79
- [28] Vassiliadis, Panos, Simitsis, Alkis, Skiadopoulou, Spiros, 2002, „Conceptual Modeling for ETL Processes“, *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP*, pp. 14 - 21