SYSTEMATISCHE ERFASSUNG, VERWALTUNG UND NUTZUNG VON DATEN AUS EXPERIMENTEN

F. Krebs, M. Willmeroth, T. Haase, P. Kaufmann, Dr. R. Glück, D. Deden, L. Brandt, M. Mayer
Deutsches Zentrum für Luft- und Raumfahrt e.V., Institut für Bauweisen und Strukturtechnologie Zentrum für Leichtbauproduktionstechnologie, Am Technologiezentrum 4, Augsburg, Deutschland

Zusammenfassung

Künstliche Intelligenz und Analytics halten immer mehr Einzug in den Wissenschaftsalltag. Gerade der Umgang mit Daten aus komplexen multi-Schritt Experimenten stellt heute noch eine Herausforderung dar. Das am DLR entwickelte integrierte Datenmanagementsystem „shepard“ [1] vereinfacht den Einstieg und die Verarbeitung dieser Daten. Im Folgenden wird kurz auf die Motivation hinter Experimentaldatenmanagement eingegangen. Danach wird die Basisarchitektur der Einzelkomponenten von shepard dargestellt. Am Beispiel von thermoplastischem AFP wird die kontextualisierte Datenerfassung verdeutlicht und abschließend ein Ausblick auf Weiterentwicklungsperspektiven gegeben.

Keywords

Forschungsdatenmanagement; Produktion; Qualitätssicherung; Thermoplastisches Fiber Placement

NOMENKLATUR

Abkürzungen

AFP	Automated Fiber Placement
DRG	Daten Referenz Generator
EC_M	EtherCAT Master [2]
EC_S	EtherCAT Slave [2]
FPZ	Fiber Placement Zelle (am DLR) [3]
FSD	KUKA Fast Send Driver: Hochgeschwindigkeitsschnittstelle für Roboterpositionsdaten
$HTTP$	Hypertext Transfer Protocol
$IDMS$	integriertes Datenmanagementsystem
$JSON$	JavaScript Object Notation
JWT	JSON Web Token
$MTLH$	Multi-Tape Laying Head
PN_C	ProfiNet Controller [4]
PN_D	ProfiNet Device [4]
$REST$	Representational State Transfer
$shepard$	storage for heterogeneous product and research data
SPS	Speicherprogrammierbare Steuerung
TPS	Tape Profile Sensor
UA	OPC UA: Open Platform Communications Unified Architecture [5]

1. MOTIVATION FÜR DAS MANAGEMENT VON EXPERIMENTALDATEN

Das heutige Forschungsumfeld ist geprägt von komplexen und dezentralisierten Experimenten in unterschiedlichen Themenfeldern. Zur wertschöpfenden Analyse der darin gewonnenen Experimentaldaten ist eine übergreifende Zusammenführung und Kontextuierung dieser Daten von essentieller Bedeutung. Geleitet durch die Vision der nahtlosen digitalen Integration unterschiedlichster Prozessketten [6] [7] wurde am Zentrum für Leichtbauproduktionstechnologie das integrierte Datenmanagementsystem „shepard“ (storage for heterogeneous product and research data) entwickelt.

Ein Schwerpunkt ist dabei die disziplinübergreifende Nutzbarmachung aller erzeugten Daten u.a. für verschiedenste Analyse- und KI-Methoden. Das skalierbare System ermöglicht eine hochflexible automatisierte Speicherung und Verknüpfung heterogener Daten (u.a. Messwerte, Simulationsergebnisse) und Metadaten entlang unterschiedlichster realer und digitaler Prozessketten. Über diesen einfachen und nachhaltigen Weg zur Ablage, zum Abruf, zur Analyse und zum Teilen der Experimental- und Forschungsdaten wird eine übergreifende Zusammenarbeit der Forschenden ermöglicht.

Neben der damit geschaffenen Basis für ein durchgängiges Forschungsdatenmanagement vom Versuch bis zur Publikation werden zahlreiche Anknüpfungspunkte für Analyticsframeworks eröffnet. Insbesondere die disziplin- und prozessübergreifende Auswertung der Daten kann an dieser Stelle neue Erkenntnisse generieren. Eine umfassende

doi: 10.25967/550315

prädiktive Instandhaltung sowie ein geschärftes Bewusstsein über den Umfang und Zustand der verfügbaren Datensätze trägt dazu bei, die Qualität der Experimentaldaten nachhaltig verbessern.

2. ARCHITEKTUR DES DATENMANAGEMENTSYSTEM

Zur Etablierung eines durchgängigen Datenmanagements für Daten aus Experimenten im Forschungsumfeld ist ein klarer Fokus auf Modularisierung und dadurch einfache Erweiterbarkeit notwendig. Durch klare Trennung von Komponenten können neue Datenquellen sowie Verarbeitungs- und Analyseketten problemlos integriert werden.
Im Folgenden werden die zentralen Komponenten des Datenmanagementsystems grundlegend beschrieben und ihre Funktion dargestellt.

2.1. Basisarchitektur

Im Forschungsumfeld werden viele verschiedene Daten von unterschiedlichen Typen behandelt. Das Spektrum reicht von einfachen Parameterlisten bis hin zu komplexen Datenstrukturen. Daneben fallen in vielen Fällen Zeitreihen sowie binäre Dateien an. Ein Datenmanagementsystem soll solche Daten effizient speichern und verarbeiten können. Aus diesem Grund wurde entschieden, einen *Polyglot Persistence* Ansatz zu verfolgen und unterschiedliche Datenbanken für die jeweiligen speziellen Datentypen und Aufgaben zu verwenden und zu kombinieren. [8] Während Verwaltungsdaten, Organisationselemente und Metadaten in der Graphdatenbank neo4j gespeichert werden, werden Zeitreihen in der Zeitreihendatenbank InfluxDB abgelegt. Strukturierte Daten in Form von JSON Strings sowie binäre Dateien werden in MongoDB gespeichert. Der Grund für diese Aufteilung ist, dass sowohl MongoDB als auch InfluxDB Methoden zum Suchen in den jeweils dort gespeicherten Daten mitbringen, wohingegen die Stärke von neo4j in der einfachen Verwaltung von Beziehungen zwischen den gespeicherten Objekten liegt.
Das Datenmanagementsystem soll von unterschiedlichen Datenquellen und Analysesystemen verwendet werden können. Aus diesem Grund werden sämtliche Funktionen über eine einheitliche *Representational State Transfer (REST)* Schnittstelle bereitgestellt. [9] Darüber hinaus können die enthaltenen Daten sowie einzelne Funktionen auf einer Weboberfläche dargestellt werden, die auf dieser REST Schnittstelle aufbaut.
Die REST Schnittstelle ist durch ein OpenAPI-Dokument beschrieben. [10] Diese Schnittstellenbeschreibung wird automatisiert aus der Software abgeleitet und ermöglicht es, automatisiert Softwarebibliotheken für verschiedene Programmiersprachen zu generieren. So wird die Einstiegshürde für Anlagenbetreiber, Datenlieferanten und -analysten deutlich verringert.

2.2. Backend

Das Backend stellt die Schnittstelle zwischen REST und den jeweiligen Datenbanken bereit. Es ist in Java implementiert und verwendet die JAX-RS Referenzimplementierung Eclipse Jersey zur Darstellung einzelner Ressourcen. Für die Bereitstellung des generierten Servlets wird ein Apache Tomcatserver verwendet.
Die Daten sind aktuell in drei verschiedenen Datenbanken vorgehalten. Dabei liegen Organisationselemente als leere Hülsen in der neo4j Datenbank, die dann durch die Nutzdaten aus der MongoDB bzw. der InfluxDB gefüllt werden können.
Die Authentifizierung/Autorisierung von Benutzern geschieht mittels JSON Web Tokens (JWT). [11] Entsprechend der OAuth2 Spezifikation werden diese Tokens durch einen Client angefragt und zur Absicherung der REST Schnittstelle verwendet. [12] Da Anlagen und andere Systeme jedoch nicht über eine Webseite, sondern per Skript auf die REST Schnittstelle zugreifen, eignet sich OAuth2 nicht für diese Anwendungsfälle. Für diesen Fall wurden sogenannte API Keys eingeführt, welche durch einen Benutzer erstellt werden können und zur Authentifizierung dieses Benutzers verwendet werden können. Im Gegensatz zu Access Tokens sind API Keys unbegrenzt gültig. Im Falle eines Verlusts kann der Benutzer den entsprechenden API Key im Backend ungültig machen.

2.3. Datenmodell und API

Angelehnt an die Empfehlungen der *Research Data Collections WG* werden die Daten in shepard in sogenannten *Collections* organisiert. [13] Collections dienen der Gruppierung untergeordneter Datenelemente und enthalten sogenannte *DataObjects*. DataObjects bilden die eigentlichen Daten ab und lassen sich hierarchisch und logisch strukturieren. So kann jedes DataObject genau ein Elternobjekt und mehrere Vorgänger, Nachfolger und Kindobjekte haben. Mithilfe der Eltern- und Kindobjekte lassen sich hierarchische Strukturen abbilden, während die Vorgänger und Nachfolger dazu dienen, temporale oder kausale Abhängigkeiten zu modellieren. Diese referenzierten Objekte müssen ebenfalls DataObjects sein.

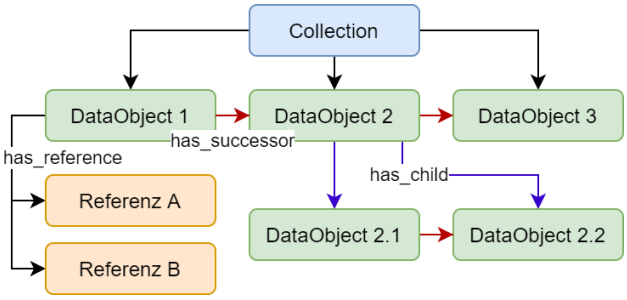

BILD 1. Schematisches Datenmodell

Nutzdaten sind in sogenannten *Containern* organisiert. Es gibt eine Containerart für jeden Datentyp. Ein Container kann mehrere Nutzdatenobjekte enthalten. Die Verknüpfung zwischen DataObjects und Nutzdaten geschieht mittels sogenannter *References*. References gehören immer zu jeweils einem DataObject und zeigen auf ein oder mehrere Nutzdatenobjekte in genau einem Container.
Das Datenmodell ist beispielhaft in Abbildung 1 dargestellt.

2.4. Frontend

Das Frontend dient hauptsächlich zur manuellen Navigation durch die abgelegten Daten. Die Kommunikation mit dem Backend erfolgt genau wie bei der automatisierten Datenablage über den OpenAPI Client. Ziel ist es dabei, möglichst genau die API abzubilden, um die manuelle Navigation ähnlich zur automatisierten Schnittstellenverwaltung zu halten. Weiterhin können die Daten auch durch Erzeugen, Bearbeiten oder Löschen von Collections, DataObjects, Datencontainern oder Referenzen strukturiert und verwaltet werden. Aus diesem Grund wurde das Frontend möglichst datenorientiert designet. Ein logischer Schritt war daher die Verwendung von TypeScript, da TypeScript im Gegensatz zu JavaScript Objektorientierung und Typsicherheit bietet. [14] [15] Weiterhin wurde entschieden, ein Frontend Frameworks zu verwenden. Die drei in Frage kommenden und untersuchten Frameworks waren Angular, React und Vue. Nach dem Vergleich der drei Frameworks mit unserem Anforderungsprofil fiel die Entscheidung schließlich für das jüngste der drei Frameworks, nämlich Vue. Anders als React ist Vue keine Ansammlung von vielen Bibliotheken, was React zwar sehr leichtgewichtig und flexibel macht, dafür aber weniger stringent in der Problemlösung. Vue ist genau wie Angular ein monolithisches Framework, wenngleich leichtgewichtiger und intuitiver nutzbar. Diese Eigenschaften gaben den Ausschlag für die Verwendung von Vue. [16]

3. THERMOPLASTISCHES IN-SITU AUTOMATED FIBER PLACEMENT

3.1. Anwendungsfallbeschreibung

Automated Fiber Placement (AFP) ist ein additiver Fertigungsprozess, bei dem ein Bauteil in einer Werkzeugform aus mehreren aufeinanderfolgenden Lagen aus carbonfaserverstärktem Kunststoff (CFK) aufgebaut wird. Jede Lage besteht aus mehreren Bändern bzw. Tapes, die einzeln oder in Gruppen abgelegt werden. Die Besonderheit des thermoplastischen in-situ AFPs ist, dass der Prozess einstufig ist und es daher keinen nachfolgenden Konsolidierungsschritt gibt. Das Ausgangsmaterial aus thermoplastischer Matrix ist bereits mit Endlosfasern aus Kohlenfaserstoff verstärkt. Während des Prozesses werden die Bänder und das darunterliegende Laminat aufgeschmolzen. Unter Druck finden Polymer-Diffusionen statt, durch die Laminat und Bänder miteinander verschmelzen. Für eine erfolgreiche Ablage sind der Konsolidierungsdruck, die Verarbeitungstemperatur und die Positionierung entscheidend. Sowohl der Druck als auch die Temperatur werden im Prozess geregelt. Die Positionierung der Bänder wird mit Hilfe eines integrierten Sensors, der Höhenprofile aufnimmt, analysiert; Details finden sich in Unterabschnitt 3.3. Zusätzlich werden bei der Ablage eine Vielzahl von Daten erzeugt und gespeichert, wie beispielsweise die Temperatur der Bänder, Zug- und Druckkräfte sowie Geschwindigkeiten. Aus ihnen lassen sich Aussagen zur Qualität des Bauteils ableiten. Eine systematische Aufnahme und Verknüpfung der Daten mit Informationen wie der zugehörigen Roboterposition erlaubt weitere Rückschlüsse auf die Position von Fehlstellen. Wegen der direkten Konsolidierung des CFKs ist diese Qualitätssicherung essentiell, um die Güte des Bauteils zu bewerten und eventuelle Reparaturen durchführen zu können.

3.2. Zelleninfrastruktur und Schnittstellen

Am ZLP existiert eine sogenannte Fiber Placement Zelle (FPZ), die speziell für die Anforderungen von AFP-Prozessen entworfen wurde. Die Hardware der FPZ besteht - aus Sicht der Datenerhebung - aus vier wesentlichen Komponenten 3:

- Einer Robotersteuerung vom Typ KUKA KRC 4, mit der ein KUKA KR-120 HA betrieben wird,
- einer Kopfsteuerung, die den eigentlichen Ablegekopf „MTLH“ der Firma AFPT sowie die integrierte Laserheizung (Laserline LDM 600) steuert und regelt,
- einer Steuerung für den Tape Profile Sensor (TPS) zur integrierten Prozessüberwachung (siehe Unterabschnitt 3.3) sowie
- einer SPS zur Gesamtsteuerung der Fertigungszelle und sicherheitstechnischen Überwachung.

Des Weiteren wird ein Industrie-PC als zentraler Leitstand eingesetzt, auf dem alle notwendigen Parameter für den eigentlichen Versuch sowie dessen Datenerfassung erhoben werden können. Um die Netzwerklast bei der Erfassung der hochfrequenten

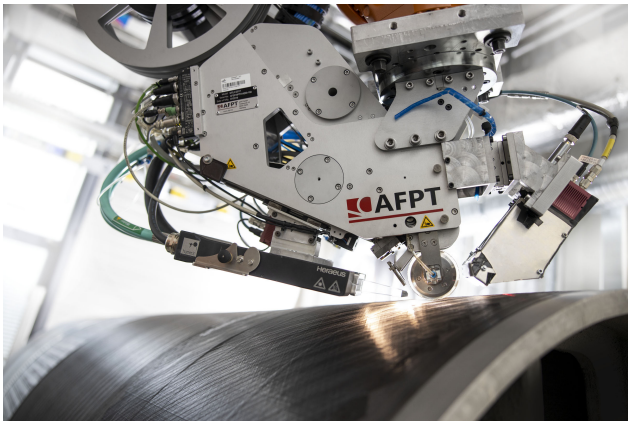

BILD 2. Thermoplastisches Tapelegen

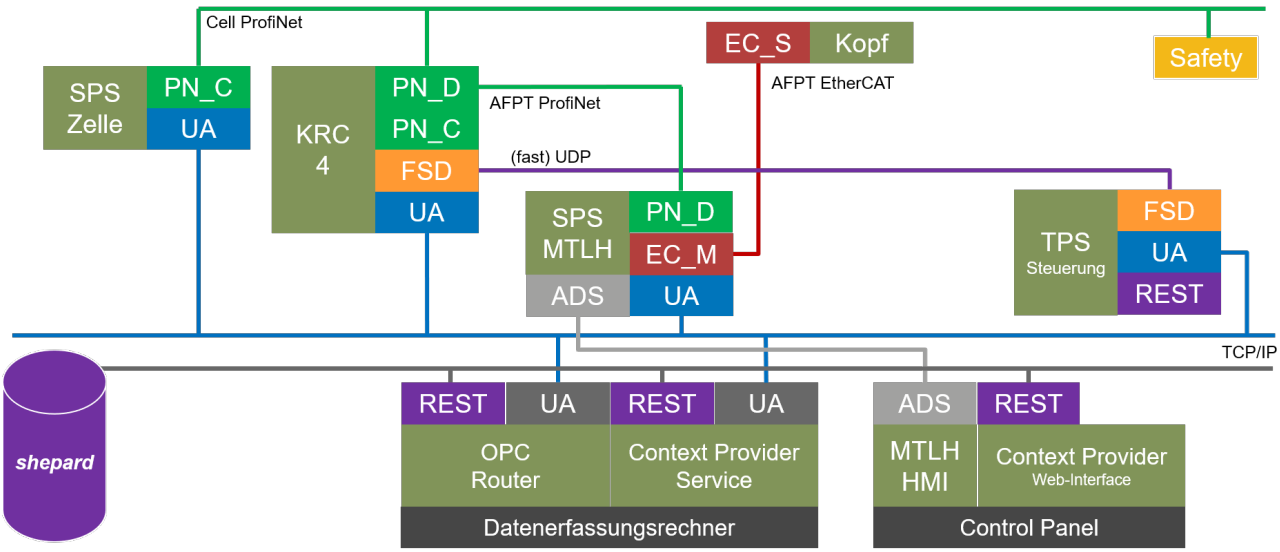

BILD 3. Netzwerk- und Schnittstellenschema FPZ

Zeitreihendaten zu verteilen und eine Beeinflussung der Prozesssteuerung zu vermeiden, wurde noch ein zusätzlicher Datenerfassungsrechner eingerichtet.
Alle Datenverbindungen ab Teilsteuerungsebene werden über klassisches (industrielles) Ethernet realisiert (siehe Abbildung 3). Im Rahmen der eigentlichen Prozesssteuerung werden klassische industrielle Feldbusse (ProfiNet [4], EtherCAT [2]) verwendet. Die eigentliche Erfassung der Zeitreihendaten erfolgt über OPC-UA [5], jedoch sind andere Datenquellen leicht integrierbar, solange ein Werkzeug die Übersetzung in entsprechende REST Aufrufe leistet.

3.3. Tape Profile Sensor

Bei der Ablage von Tapes kann es zu Abweichungen kommen. Liegen die Tracks (eine simultan abgelegte Gruppe von Bändern) nicht genau auf Stoß, entsteht ein sogenannter *gap*. Diese Abweichung kann mittels des eigenentwickelten TPSs gemessen werden [17]. Dazu wird eine Laserlinie quer zur Ablegerichtung auf die Tapes gerichtet. Eine Kamera zeichnet das Höhenprofil und damit auch den gap auf. Der TPS befindet sich fest montiert am Endeffektor und akquiriert Daten während des eigentlichen Tapelegeprozesses. Neben gaps können auch *overlaps* sowie *rising* und *falling edges* detektiert werden. Eine Erkennung von weiteren Merkmalen und Anomalien [18] wie *early/late cuts*, *foreign objects* oder *tape splices* ist ebenfalls möglich, erfordert jedoch noch eine Implementierung der entsprechenden Erkennungsalgorithmen.
Die Auswertung findet parallel zum Prozess statt. Eine Zusammenfassung der Ergebnisse wird mittels OPC-UA bereits während der Ablage zur Verfügung gestellt und in der Zeitreihendatenbank gesichert. Am Ende eines jeden Tracks werden alle erfassten Daten (Bilder, Punktwolkeninformationen, siehe Abbildung 5) mittels API Calls in shepard übertragen. Da shepard die entsprechende Prozessorganisation übernimmt, können Daten auf einfache Weise den richtigen Prozessschritten zugeordnet werden.

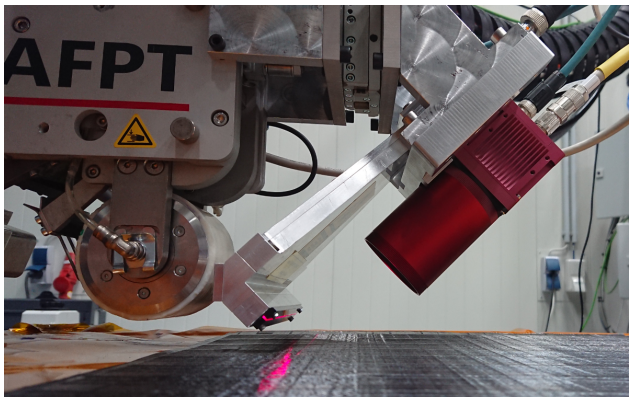

BILD 4. Tape Profile Sensor misst kurz nach der Anpressrolle während des Tapelegeprozesses

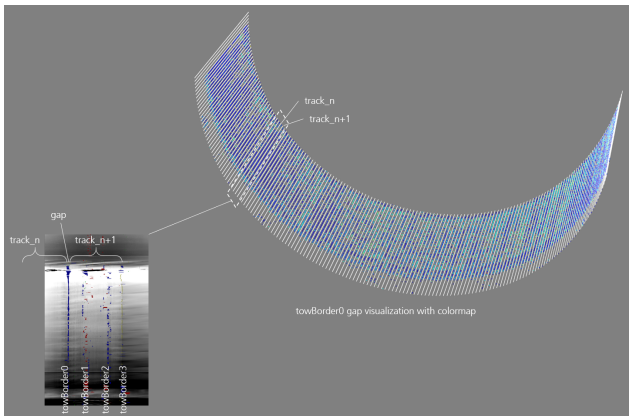

BILD 5. Exemplarische Visualisierung von Gaps zwischen den gelegten Tracks

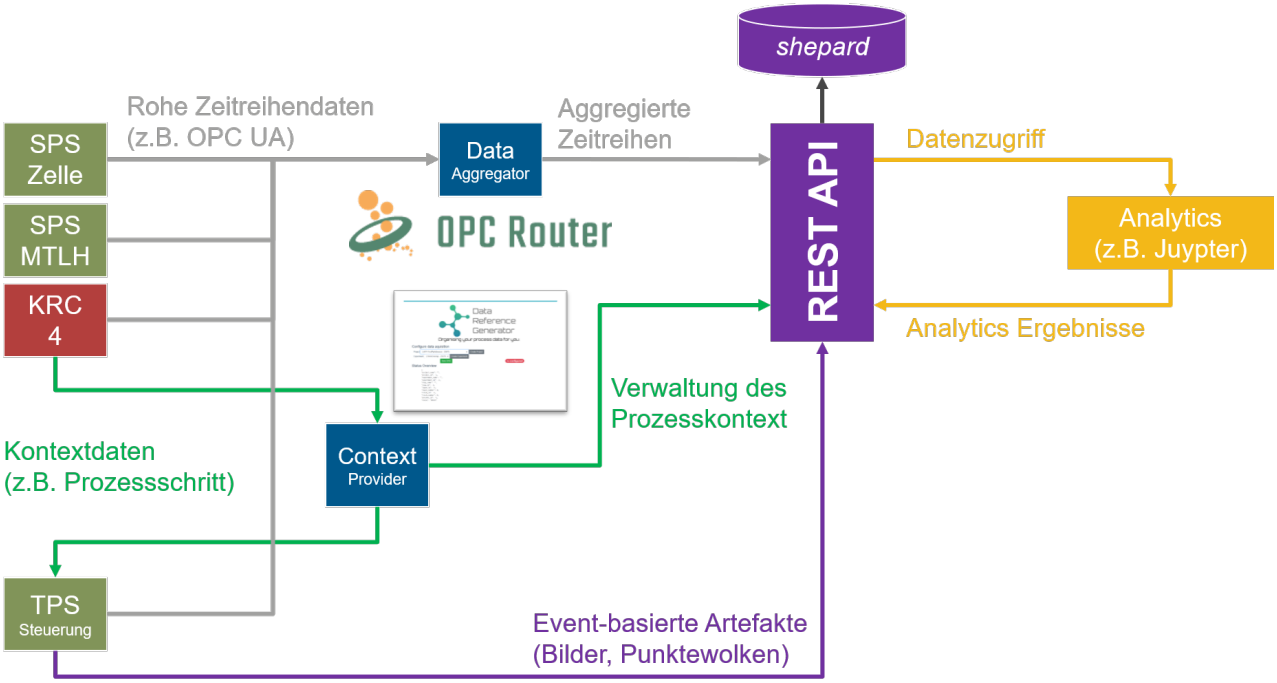

BILD 6. Schematischer Datenfluss vom Experimentalaufbau zum Datenmanagementsystem

4. DATENFLUSS UND KONTEXTUALISIERUNG

Um aus Rohdaten weitergehende Informationen abzuleiten, müssen diese in einen *Kontext* gesetzt werden. Kontext bezeichnet dabei die umgebenden Bedingungen, unter denen die Rohdaten gewonnen wurden. Am Beispiel des thermoplastischen Tapelegens (siehe Abschnitt 3) heißt das, dass beispielsweise die Umweltbedingungen (Raumtemperatur und Luftdruck) mit erfasst werden. Noch wichtiger ist aber der logische Zusammenhang, in dem die Daten gewonnen wurden. Dabei werden in diesem Beispiel die Umgebungsinformationen hinsichtlich des Lagenaufbaus (Lage und Track) des Bauteils (z.B. Experimentlaufnummer) erfasst. Meist verfügt nur ein Element des Experimentalaufbaus über diese Informationen und muss diese für die weitere Datenerfassung zur Verfügung stellen. Hinsichtlich der Rohdaten werden zwei Arten von Daten erfasst und unterschiedlich behandelt. Einerseits werden Zeitreihendaten (in hoher Frequenz) erfasst, andererseits können *spontan entstehende*, event-orientierte Artefakte dem aktuellen Kontext zugeordnet werden. Im Folgenden wird ein exemplarischer Datenfluss vom Experimentalaufbau zum Datenmanagementsystem dargestellt (siehe Abbildung 6). Wie in Unterabschnitt 4.1 noch beschrieben wird, werden Zeitreihen ständig erfasst. Dabei werden die Daten zusammengefasst und an die API von shepard übergeben.
Der Context Provider beobachtet den Versuchsverlauf anhand einer Führungskomponente. Diese hat Informationen über den aktuellen Prozess- und Versuchszustand und kann dadurch den Prozesskontext auf shepard verwalten. Gleichzeitig können diese Informationen auch weiteren Komponenten (wie z.B. dem TPS) zur Verfügung gestellt werden. Diese Systeme können dann event-orientierte Daten (z.B. Bilder, die nur zu bestimmten Prozesszeitpunkten aussagekräftig sind) an die entsprechenden Organisationselemente anfügen.

4.1. Erfassung von Zeitreihendaten

Die Erfassung der Zeitreihendaten erfolgt dauerhaft. Das heißt, auch wenn kein Prozesskontext besteht - wie z.B. bei einem kurzen Test des Systems - werden die Zeitreihendaten erfasst und in shepard gesichert. Dadurch wird sichergestellt, dass bei interessanten Ergebnissen diese Daten trotzdem vorliegen und auch nach der Ausführung noch *manuell* kontextualisiert werden können. Hierfür wird die kommerzielle Software OPC-Router [19] eingesetzt, die eine einfache Konfiguration der zu erfassenden Datenelemente erlaubt. In OPC Router können dabei unterschiedlichste industriell standardisierte Input- (wie z.B. OPC UA [5]) und Output-Ziele bzw. -Protokolle (z.B. REST mittels HTTP [9] auf der Basis von OpenAPI [10]) definiert werden, um Daten aus Quell- in Zielsysteme zu überführen. Im Anwendungsfall des thermoplastischen Tapelegens wurde eine parallele, zyklische Erfassung von mehr als 100 Zeitreihenwerten alle 25 ms konfiguriert. Zudem beinhaltet OPC Router die Möglichkeit zur Überwachung der Datenerfassung.

4.2. Generierung von kontextualisierten Datenreferenzen

Zur Generierung und Verwaltung des Prozesskontexts wurde der *Daten-Referenz-Generator* (DRG) entwickelt. Dieses Werkzeug beobachtet die Führungskomponente, die über die Informationen zum aktuellen Prozessverlauf verfügt (im Beispiel die

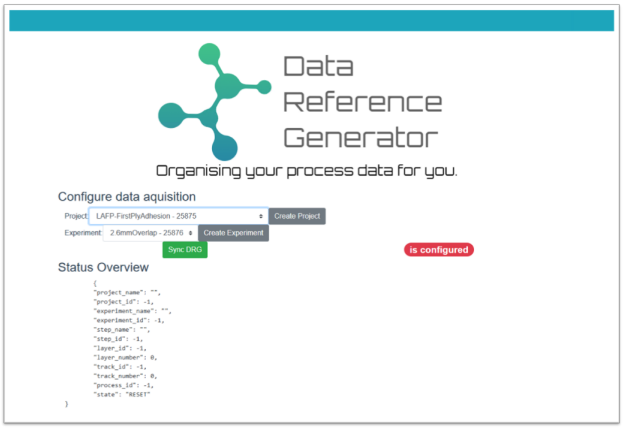

BILD 7. Weboberfläche des Daten-Referenz-Generator

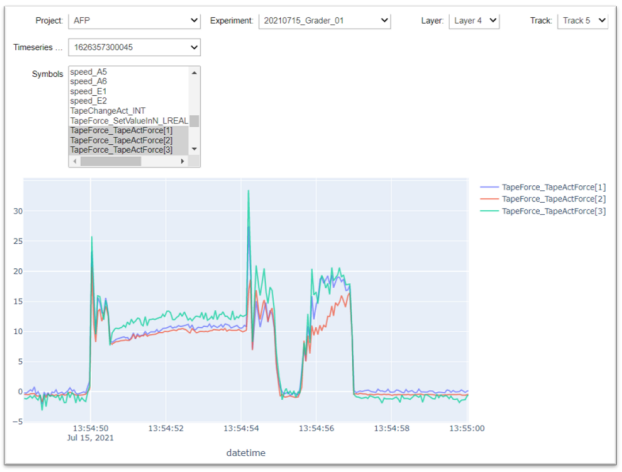

BILD 8. Interaktives Plotting Werkzeug in Jupyter Notebook

Robotersteuerung „KRC 4“, siehe Abbildung 6). Dabei erzeugt der DRG die entsprechenden Datenelemente (siehe Unterabschnitt 2.3) und stellt den aktuellen Kontext in Form einer „Prozess-ID“ zur Verfügung. Trotz aller automatisierter Ableitung des Prozesskontexts verfügt keine Komponente über organisatorische Informationen hinsichtlich des bearbeitenden Projekts und des Titels des aktuellen Experiments. Zu diesem Zweck verfügt der DRG über eine Weboberfläche, die dem Nutzer zu Beginn des Experiments ermöglicht, diese Daten einfach zu erfassen (siehe Abbildung 7). Zudem verfügt der DRG über eine Kommentarfunktion, die es ermöglicht, im Prozessverlauf Kommentare zu möglichen Anomalien und Unregelmäßigkeiten zu erfassen.

4.3. Erfassung von event-orientierten Daten

Durch die generierte „Prozess-ID“ aus Unterabschnitt 4.2 steht weiteren Datenquellen die Möglichkeit offen, Daten event-gestützt an das Datenmanagementsystem zu übertragen. Events oder Ereignisse sind dabei beispielsweise das Erreichen eines Track-Endes oder auch der Abbruch eines Prozesses - im Prinzip lässt sich dabei jedes *maschinell erkennbare* Ereignis nutzen. Die Funktion der kontextualisierten Erfassung lässt dabei zu, dass z.B. bei Erreichen des Track-Endes die erfassten Daten des TPS (Unterabschnitt 3.3) (in diesem Fall Profildaten, Highspeed-Positionsdaten und Punktewolkeninformationen) asynchron abgelegt werden können.
Hierdurch können heterogene Daten aus unterschiedlichen Quellen einem konkreten Kontext zugeordnet werden, die dann gemeinsam „im Kontext“ analysiert werden. Im Folgenden zeigen wir zwei mögliche kontextualisierte Auswertungen auf Basis der Daten aus dem Anwendungsfall.

4.4. Datenanalyse

Die Möglichkeiten zur direkten Anbindung von Analyseapplikationen über die REST API (siehe Unterabschnitt 2.3) werden anhand von Jupyter Notebooks demonstriert. [20]

Zur Analyse der erfassten Zeitreihen wurde ein interaktives Plotting Werkzeug implementiert, das das einfache strukturierte Sichten von Datensätzen auf Basis des Prozesskontext (in diesem Fall ein einzelner „Track“) erlaubt (siehe Abbildung 8).
Aber auch im Kontext der kompletten „Lage“ lassen sich die erfassten Zeitreihendaten mittels Python-Skripten oder beliebigen weiteren Tools verarbeiten und systematisch analysieren. Anhand der im Prozess erfassten Positionsdaten lassen sich Temperaturverteilungen - im Gegensatz zu oben - spatial darstellen (siehe Abbildung 9). Weiterführend können aus diesen Daten beispielsweise 3D-Oberflächen generiert werden und mit weiteren Prozess- oder Qualitätssicherungsdaten aus dem System in Beziehung gesetzt werden oder auch mit dem „Soll“ aus dem Engineering verglichen werden. Beispielsweise lassen sich während oder nach dem Prozess erkannte Anomalien mit Daten des TPSs (Unterabschnitt 3.3) und den erfassten Temperatur- oder Druckwerten des Prozesses in Relation bringen.

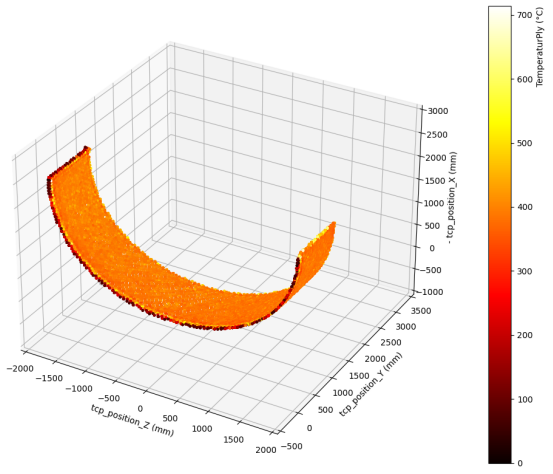

BILD 9. Erster Ausblick für 3D-Darstellungen mittels Python

Dieser Datenvergleich kann stark vom gezielten Einsatz von KI-Methoden profitieren und weitere Erkenntnisse zu noch unbekannten Prozesszusammenhängen liefern. Erste Ansätze dazu bieten „exploratory Data Analysis“ (EDA) Werkzeuge wie „sweetviz“ [21] oder „pandas-profiling“ [22] zur automatischen Korrelationsanalyse. Auf Basis dieser Analysen können weitere Features extrahiert bzw. modelliert werden und damit schnell zu KI-Anwendungen weiterentwickelt werden.

5. ZUSAMMENFASSUNG UND AUSBLICK

Durch die Entwicklung und den prototypischen Einsatz in unterschiedlichsten Disziplinen (von Flugexperimenten, über Produktionstechnik bis hin zu virtuellen Simulationsworkflows) kann das System bereits sehr viele Domänen im Kontext der Forschung insbesondere des Luft- und Raumfahrtsektors abdecken. Eine detaillierte Rückmeldung aus den bisherigen Einsatzgebieten erlaubt die kontinuierliche Steigerung der Systemqualität und -performance. Die bereits bestehenden und in diesem Dokument beschriebenen Funktionalitäten werden im Zuge dessen erweitert und verbessert.
Eine Einbindung weiterer Datenquellen beispielsweise zur Code- und Versionsverwaltung über Git wird konzeptionell vorbereitet. Auf diese Weise wird eine direkte Verknüpfung von Datensätzen und Experimenten mit spezifischen Software- und Entwicklungsständen ermöglicht. Dies trägt insbesondere zur Erhöhung der Nachvollziehbarkeit und Reproduzierbarkeit der Experimentaldaten sowie der durch Analyse gewonnenen Erkenntnisse bei.
Perspektivisch wird eine Ergänzung im CAD-Umfeld durch ein Produktdatenmanagementsystem erfolgen. Analog zur Referenzierung der Datensätze mit spezifischen Softwareversionen können so relevante CAD-Informationen nachvollziehbar bereitgestellt werden.
Um die Zugänglichkeit der im System gespeicherten Datensätze zu erhöhen, werden weitere Such- und Filterfunktionen für einen vereinfachten Datenzugriff vorbereitet. Die strukturierte Suche in Organisationselementen sowie der Zugriff auf interne Filterfunktionen von Zeitreihen- und Dokumenten-Datenbanken erleichtert die Auswahl der Datensätze aus einem spezifischen Kontext zur gezielten Analyse. In diesem Kontext ist ein solides Rechtemanagement von großer Bedeutung. Zum Zeitpunkt der Erstellung des Dokuments finden daher bereits Arbeiten zur Zugriffs- und Rechteverwaltung innerhalb des Systems statt.
Über die Nutzung des Systems sollen zukünftig u.a. festgestellte Anomalien von Bauteilen in der Engineeringkette zurückverfolgt und das Auftreten unerwünschter Effekte sowie deren Ursachen anhand der Daten vorhergehender Arbeitsschritte näher spezifiziert und erklärt werden können. Durch die Auswertung der vorhandenen Daten sollen nötige Modifikationen in der Prozesskette sowie in den Prozessen identifiziert werden, um diesen Anomalien zu begegnen oder sie bereits vor ihrer Entstehung

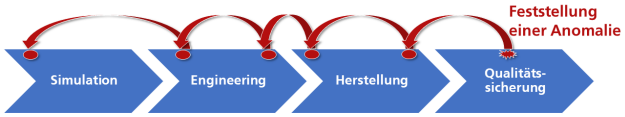

BILD 10. Auszug einer Engineeringkette aus der Produktion

zu verhindern. Weitergehend kann eine Vorhersage der zu erwartenden Bauteilqualität aus den im System erfassten Daten und Metadaten vorheriger Bauteile bereits während der Produktion ermöglicht werden. Auch im Bezug auf eine vorausschauende Wartung beteiligter Maschinen und Komponenten bieten vollumfängliche Datensätze viele Vorteile.
Mit der Veröffentlichung unter der Apache 2.0 Lizenz steht das System shepard inzwischen einer breiten Community zur Nutzung und Weiterentwicklung zur Verfügung.

Kontaktadresse:

florian.krebs@dlr.de

Literatur

[1] T. Haase, Dr. R. Glück, P. Kaufmann und M. Willmeroth. shepard - storage for heterogeneous product and research data, July 2021. DOI: 10.5281/zenodo.5091603.

[2] EtherCAT Technology Group. EtherCAT - Der Ethernet-Feldbus. https://www.ethercat.org/de/technology.html. (Accessed on 07/08/2021).

[3] DLR - Institut für Bauweisen und Strukturtechnologie. Fibre Placement Zelle (FPZ). https://www.dlr.de/bt/desktopdefault.aspx/tabid-2488/3746_read-59647/#/gallery/34212. (Accessed on 07/08/2021).

[4] PROFIBUS Nutzerorganisation e.V. (PNO). PROFINET. https://www.profibus.com/technology/profinet. (Accessed on 07/08/2021).

[5] OPC Foundation. Unified Architecture - OPC Foundation. https://opcfoundation.org/about/opc-technologies/opc-ua/. (Accessed on 07/07/2021).

[6] C. Frommel, F. Krebs, T. Haase, M. Vistein, A. Schuster, L. Larsen , M. Körber, M. Malecha und M. Kupke. Automated manufacturing of large composites utilizing a process orchestration system. *Procedia Manufacturing*, 51:470–477, 2020. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021). DOI: 10.1016/j.promfg.2020.10.066.

[7] F. Krebs. Towards plug and work opc ua as middleware of modern automation systems. In *ISR 2018; 50th International Symposium on Robotics*, pages 1–6, 2018.

[8] F. Gessert, N. Ritter. Polyglot persistence. *Datenbank-Spektrum*, 15(3):229–233, November 2015. DOI: 10.1007/s13222-015-0194-1.

[9] W3C. REST - Semantic Web Standards. https://www.w3.org/2001/sw/wiki/REST. (Accessed on 07/07/2021).

[10] The Linux Foundation. OpenAPI Specification v3.1.0. https://spec.openapis.org/oas/latest.html. (Accessed on 07/07/2021).

[11] M. Jones, J. Bradley und N. Sakimura. JSON Web Token (JWT). RFC 7519, May 2015. DOI: 10.17487/RFC7519.

[12] D. Hardt. The OAuth 2.0 Authorization Framework. RFC 6749, October 2012. DOI: 10.17487/RFC6749.

[13] T. Weigel, B. Almas, F. Baumgardt, T. Zastrow, U. Schwardmann, M. Hellström, J. Quinteros, D.Fleischer. Recommendation on research data collections. page 44, 10 2017. DOI: 10.15497/RDA00022.

[14] TypeScript Documentation. https://www.typescriptlang.org/docs/, 2021. (Accessed on 26/07/2021).

[15] G. Bierman, M. Abadi, M. Torgersen. Understanding typescript. In *European Conference on Object-Oriented Programming*, pages 257–281. Springer, 2014.

[16] E. Wohlgethan. Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue.js, 2018. DOI: 20.500.12738/8417.

[17] A. Schuster, M. Mayer, M. Willmeroth, L. Brandt und M. Kupke. Inline quality control for thermoplastic automated fibre placement. *Procedia Manufacturing*, 51:505–511, 2020. 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021). DOI: 10.1016/j.promfg.2020.10.071.

[18] R. Harik, C. Saidy, S. J. Williams, Z. Gurdal,B. Grimsley. Automated Fiber Placement Defect Identity Cards: Cause, Anticipation, Existence, Significance, and Progression. In *SAMPE 2018*, Long Beach, CA, May 2018.

[19] OPC Router: Die Industrie 4.0 Software zur Systemintegration. https://opc-router.de/. (Accessed on 07/07/2021).

[20] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, C. Willing und Jupyter development team. Jupyter notebooks - a publishing format for reproducible computational workflows. In F. Loizides and B. Scmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90. IOS Press, 2016. DOI: 10.3233/978-1-61499-649-1-87.

[21] F. Bertrand. sweetviz: Visualize and compare datasets, target values and associations, with one line of code. https://github.com/fbdesignpro/sweetviz. (Accessed on 07/30/2021).

[22] S. Brugman. pandas-profiling: Exploratory Data Analysis for Python. https://github.com/pandas-profiling/pandas-profiling, 2019. (Accessed on 07/30/2021).