



Digitale Zwillinge – viele Tools, ein Überblick

Digitale Zwillinge gelten als Schlüsseltechnik in der vernetzten Industrieproduktion. Als virtuelle Abbilder von Maschinen machen sie nicht nur die Datenlandschaft übersichtlicher. Auswahl an Tools gibt es reichlich.

Von Daniel Lehner

■ Das Konzept digitaler Zwillinge entwickelt sich in den letzten Jahren immer mehr zur Schlüsseltechnik zur effizienten Verwertung von Daten, die vernetzte physische Geräte generieren. Das zeigt nicht zuletzt die Prominenz des Themas auf Branchentreffen wie der diesjährigen Hannover Messe (siehe Vorbericht auf Seite 21).

Über die in dieser Übersicht vorgestellten Tools bieten digitale Zwillinge eine einheitliche Schnittstelle, um mit physischen Geräten zu kommunizieren und eine Simulation des Verhaltens der Geräte zu steuern (siehe Abbildung 1). Die Zwillingsservices ermöglichen damit Optimierungen der physischen Systeme. Dafür benötigen sie erstens Daten vom

laufenden System und zweitens Interaktionsmöglichkeiten, um das Verhalten des Systems aktiv zu beeinflussen. Zusätzlich kann auch eine Simulation das System abbilden. Diese Simulation spiegelt das mögliche Verhalten des echten Systems und ist funktional mit ihm identisch – sie liefert dieselben Daten und bietet dieselben Interaktionsmöglichkeiten. Die virtuelle Simulation hat jedoch einige Vorteile: Weil sie rein am Computer abläuft, können bei Tests keine Geräte kaputtgehen. Zudem laufen Berechnungen wesentlich schneller als die Ausführung in einer realen Umgebung. So erlaubt eine Simulation das schnelle Ausprobieren verschiedener Szenarien in einem sicheren Umfeld.

Die vielen am Markt vertretenen Industriergeräte und Simulationen bieten jedoch verschiedene Interfaces an, was ihren gemeinsamen Einsatz erschwert. Abhilfe schafft ein zentrales Digital-Twin-Interface: Mit ihm können Mehrwertdienste, Digital-Twin-Services (DT-Services), die Daten und Interaktionsmöglichkeiten der Geräte des laufenden Systems und der verschiedenen Simulationen über eine einheitliche Schnittstelle konsumieren.

Abbildung 1 zeigt das Zusammenspiel anhand eines Beispiels. Hier besteht das laufende System aus einem Greifarmroboter, der verschiedene Fertigungsteile von unterschiedlichen Förderbändern aufnimmt und auf ein neues Förderband zum Weitertransport ablegt. Die Roboter und Förderbänder sind ans Internet angebunden und liefern regelmäßig Informationen über Parameter wie Energieverbrauch, Geschwindigkeit und Auslastung des Förderbandes oder zur Position des Robotergriffarms. Außerdem bieten die Geräte verschiedene Interaktionsmöglichkeiten per Internet wie das Stoppen von Förderbändern bei Engpässen oder das Greifen eines Teils durch einen bestimmten Roboter an.

Verschiedene Softwareservices ermöglichen auf Basis der Gerätedaten und Interaktionsmöglichkeiten Zusatzfunktionen. Während ein Dashboard zum Beispiel die Entwicklung verschiedener Parameterwerte wie Geschwindigkeit oder den Energieverbrauch eines Förderbandes überwacht, erlaubt ein Planungsservice die aktive Steuerung der Geräte zur fortlaufenden Verbesserung des Produktionsprozesses. Durch Vergleiche von Daten aus der Simulation mit dem laufenden System kann ein Predictive-Maintenance-Service Fehler erkennen, noch bevor sie in der Realität auftreten, und so frühzeitig gegensteuern.

TRACT

- ▶ Digitale Zwillinge sind eine Kernsoftwarekomponente der Industrie 4.0. Sie haben eine Reihe von Vorteilen und bieten eine einheitliche Schnittstelle sowohl zum physischen System als auch zur einer virtuellen Softwareversion des Geräts. So ermöglichen sie unter anderem die Simulation von Aktionen und Predictive Maintenance.
- ▶ Ein System für digitale Zwillinge besteht gewöhnlich aus drei Komponenten: dem zentralen Digital-Twin-Interface, einem verbindenden IoT-Hub und häufig einem Simulationstool. Alle drei Bereiche haben eigene Tools, die es in Einklang zu bringen gilt.
- ▶ Die derzeitige Marktsituation ist unübersichtlich, kein Anbieter bietet in allen Teilbereichen vollständige Softwareprodukte an. Das eigene Toolpaket sollte man daher sorgfältig zusammenzustellen.

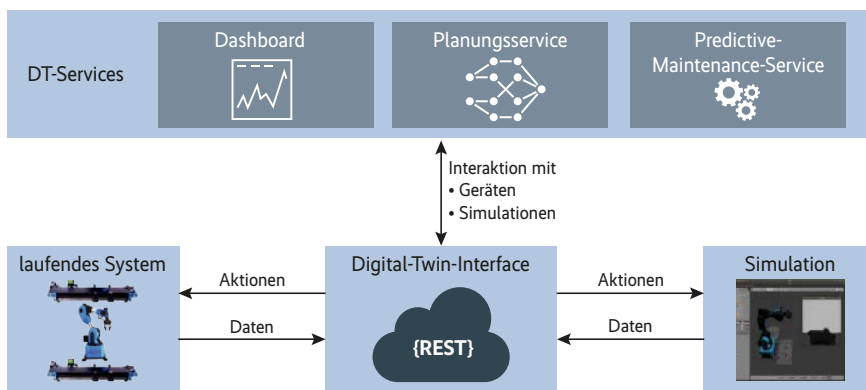
Die einzelnen Services direkt an das laufende System oder dessen Simulation anzuknüpfen wäre aber sehr aufwendig. In Produktionshallen befinden sich Geräte verschiedener Hersteller mit unterschiedlichen Datenaustauschformaten und Interaktionsschnittstellen. Die Simulationen, die das Verhalten dieser Geräte nachstellen, sind ebenfalls in verschiedenen Tools mit unterschiedlichen Schnittstellen abgebildet. Entwicklungsteams müssen bei jedem einzelnen Service alle diese Schnittstellen korrekt interpretieren und kombinieren. Dieser Aufwand ist jedoch oft so groß, dass er den potenziellen Nutzen des entwickelten Service übersteigt.

Hier kommt der digitale Zwilling ins Spiel, der über das Digital-Twin-Interface eine einheitliche Schnittstelle zum Auslesen von Daten und zum Setzen von Interaktionen von Geräten im laufenden System oder deren Simulationen ermöglicht. Ein Digitaler-Zwilling-System wie in Abbildung 1 von Grund auf neu zu etablieren ist jedoch ineffizient. Der Einsatz von Digital-Twin-Interfaces und die Verknüpfung des Interface mit dem laufenden System oder einer Simulation umfassen repetitive Aufgaben, die für jedes System gleich sind. Deshalb haben viele Softwareanbieter Tools auf den Markt gebracht, die das Erstellen digitaler Zwillinge vereinfachen. Die Auswahl der richtigen Tools ist jedoch schwierig, da digitale Zwillinge das Zusammenspiel verschiedener Bereiche erfordern. Zusätzlich bietet jeder Bereich eine ganze Reihe verschiedener Tools mit unterschiedlichen Stärken und Schwächen, was die Auswahl weiter erschwert (siehe Abbildung 2).

Dieser Artikel bringt einen praxisorientierten Vergleich der Tools für die einzelnen Bereiche und zeigt offene Herausforderungen bei der Zusammensetzung der Tools auf. Wie das Zusammenspiel verschiedener Tools bestenfalls funktionieren kann, zeigt ein früherer iX-Artikel (siehe „Zweieilige Unterstützung“ in iX 5/2022, S. 140).

Toolüberblick: Digital-Twin-Interface

Das Herzstück eines Digital-Twin-Systems ist das Digital-Twin-Interface (DT-Interface). Es ermöglicht den einheitlichen Zugriff auf die Geräte und Simulationen, meist über die Cloud. Deshalb ist es naheliegend, dass vor allem die großen Cloud-Anbieter das Portfolio ihrer bestehenden Plattformen um ein DT-Interface erweitert haben. Während Azure mit dem Azure Digital Twins Service einer



Der exemplarische Aufbau einer Digital-Twin-Architektur verdeutlicht die Vorteile der Technik: Sie ermöglicht den einheitlichen und herstellerunabhängigen Zugriff auf die Systeme der industriellen Produktion (Abb. 1).

der ersten Anbieter war, hat AWS erst vor Kurzem mit dem TwinMaker nachgezogen, Google bietet aktuell keine Digital-Twins-Software an. Neben den Cloud-Riesen sind auch andere große Player aus dem Softwarebereich auf den Digital-Twins-Zug aufgesprungen. In der Praxis haben sich vor allem die Dienste von IBM und der Bosch Semantic Stack durchgesetzt. IBM überzeugt mit der einfachen Anbindung der Watson Services an physische Geräte, der Bosch-Dienst mit der Digital Twin Registry als Kernstück und der Unterstützung der offenen Standards Asset Administration Shell (AAS) und Eclipse Semantic Modeling Framework. Die AAS bietet ein einheitliches Format zur Definition von Informationen eines physischen Systems in einem digitalen Zwilling.

Als Open-Source-Alternative zu diesen Anbietern hat sich über die letzten Jahre das Digital-Twin-Projekt der Industrial Digital Twin Association (IDTA) durchgesetzt, das unter dem Dach der Eclipse Foundation entwickelt wird. Dieses Projekt bietet eine offene Implementierung des AAS-Standards in Java und kann über Docker-Images selbst gehostet werden. Die Tabelle „Marktübersicht: Digital-Twin-Interfaces“ gibt einen Überblick, welche Aufgaben die Interfaces konkret beim Erstellen digitaler Zwillinge übernehmen und wie sich die Angebote der einzelnen Hersteller unterscheiden.



Die von iX vorgestellten Tools sitzen an den neuralgischen Punkten der digitalen Zwillinge. Ein Interface wird wie ein Hub immer benötigt, ein Simulationstool ist optional, aber häufig sinnvoll (Abb. 2).

Um mit einem DT-Interface digitale Zwillinge zu erstellen, zu ändern oder auch wieder zu löschen, sind entsprechende Funktionen im Tool nötig. Dabei ist vor allem ein grafisches Interface hilfreich. Azure bietet hier mit seinem Azure Digital Twins Explorer bereits ein sehr ausgereiftes Open-Source-Tool, während Eclipse sich auf Drittsoftware in Form bestehender Editoren des AAS-Standards verlässt. Bei anderen Anbietern müssen Anwenderinnen für diese Funktionen ein REST-Interface verwenden, das alle untersuchten Tools besitzen.

Nur durch das Erstellen digitaler Zwillinge ist jedoch noch nicht viel erreicht. Um die in der Einleitung beschriebenen Anwendungen zu realisieren und die Zwillinge mit Leben zu füllen, benötigt das DT-Interface eine Verknüpfung mit einem laufenden System oder einer Simulation des Systems. Konkret macht das Interface dabei folgende zentrale Funktionen möglich:

- Datensynchronisation: Anwenderinnen und Anwender definieren im Zwilling verschiedene Attribute, das laufende System oder die Simulation sendet anschließend unentwegt den aktuellen Attributswert an den Zwilling. Über das Interface kann dieser aktuelle Wert ausgelesen und auch verändert werden. Eine Änderung des Wertes bewirkt dabei eine entsprechende Anpassung im laufenden System beziehungsweise in der Simulation.

- **Ausführen von Operationen:** Um das System oder die Simulation aktiv zu beeinflussen, muss das Interface auch eine Möglichkeit bieten, bestimmte Operation wie das Starten oder Stoppen eines Geräts auszuführen. Für die individuelle Umsetzung solcher Features bieten IBM und Bosch nur generelle Beratungsleistungen, aber keine weitere toolseitige Softwareunterstützung. Azure und Eclipse geben konkrete Standards vor, die die entsprechende Controllersoftware im physischen System für eine Kommunikation mit dem Zwilling implementieren muss. Azure erleichtert diese Kommunikation darüber hinaus über entsprechenden SDK-Support. Zusätzlich erlaubt Azure mit seinen plattformspezifischen Event Routes, Event Hubs und Event Grids automatisierte Anknüpfungen an den später im Artikel beschriebenen Azure IoT Hub. Für die Anbindung an eine Simulation bieten sowohl Eclipse als auch Bosch Toolunterstützung an, indem sie eine Definition einer solchen Verknüpfung innerhalb eines Zwillings explizit vorsehen. Bei IBM erfolgt dies über sogenannte Aspekte, bei Eclipse gibt es sogar eine Vorlage für die Modellierung eines entsprechenden Submodels für Simulationen.

Durch eine solche Datensynchronisation können nun verschiedene Mehrwertdienste Querys am System ausführen beziehungsweise auf Basis der von den Querys retournierten Daten eine Liste an Operationen zur Ausführung am laufenden System ermitteln. Um solche Querys noch aussagekräftiger zu machen und sinnvolle Use Cases wie Machine Learning oder das Abbilden von Prozessen, die über die Grenzen einzelner Geräte hinausgehen, zu ermöglichen, benötigt das DT-Interface außerdem folgende Features:

- **Speicherung von Histories:** Um aus vergangenen Daten zu lernen, muss neben dem Abruf aktueller Attributswerte auch eine Abfrage vergangener Werte über das DT-Interface möglich sein. Alle untersuchten Tools, mit Ausnahme des AWS TwinMakers, ermöglichen mittlerweile solche Zeitabfragen. Die technische Umsetzung erfolgt dabei entweder über anbieterspezifische Tools wie den Azure Data Explorer oder über relationale Datenbanken. Zeitseriendatenbanken sind bei den meisten Anbietern aber nicht standardmäßig unterstützt.
- **Verknüpfungen zwischen Zwillingen:** Die DT-Interfaces von Azure, Eclipse,

AWS und Bosch erlauben es, Verknüpfungen zwischen Zwillingen abzubilden. Keiner der Anbieter verrät uns jedoch, wie diese Verknüpfungen technisch in Code aussehen.

Beim Vergleich von Digital-Twins-Software kommt es aber nicht nur auf die Funktionen an. Vor allem verschiedene nicht funktionale Aspekte entscheiden, wie viel Aufwand die Erstellung und Wartung der digitalen Zwillinge über die Zeit kostet. Ein wesentlicher Faktor ist dabei die Wiederverwendbarkeit. Einmal definierte Informationen sollen wenn möglich nicht an anderer Stelle neu definiert werden müssen. Eine solche Mehrfachdefinition führt zum einen zu mehr Aufwand bei der Erstellung, vor allem aber zu möglichen Inkonsistenzen bei nachträglichen Änderungen im Rahmen der Systemwartung.

Die gängigste Art der Wiederverwendung sind Zwillingstypen als vorgefertigte Schablonen für das Erstellen neuer Zwillinge. Erstellen Anwenderinnen nun einen neuen Zwilling, verwenden sie einen bestehenden Zwillingstyp als Ausgangsbasis, der bereits relevante Operationen und Datenschemata vorgibt. Einzig die Eclipse-Software lässt solche Schablonen bisher vermissen. Da dieses

Marktübersicht: Digital-Twin-Interfaces

Produkt	Azure Digital Twins	TwinMaker	Ditto	Bosch Semantic Stack	Maximo Application Suite (MAS) & Engineering Lifecycle Management (ELM)
Hersteller	Microsoft	AWS	Eclipse	Bosch	IBM
grafische Oberfläche	ADT Explorer	–	Open-Source-Editoren	–	k. A.
Support für Anbindung an laufendes System	Standard + Toolunterstützung	Data Connectors	Standard	– (Consulting notwendig)	– (Consulting notwendig)
Support für Anbindung an Simulation	–	Data Connectors	eigenes Submodell	eigener Aspekt	–
Programmiersprachenanbindung	generisch (REST) + .NET-/Java-/JavaScript-/Python-/Go-SDKs	generisch (REST)	generisch (REST)	ESMF-Integration + Java-/JavaScript-/Python-SDKs	k. A.
Speicherung von Histories	✓ (Azure Data Explorer)	✓ (über weitere AWS-Dienste)	✓ (auf Modellebene)	✓	✓ (MAS: relationale Datenbank; ELM: Versions-Histories)
Verknüpfungen zwischen Zwillingen (Modell / Code)	✓ / –	✓ / –	✓ / –	✓ / ✓	k. A.
Portabilität	kein Support	Unterstützung durch AWS-SDK	möglich für andere Tools, die AAS unterstützen	standardisierte Zwillingsdefinitionen (ESMF und AAS)	kein Support
Interoperabilität zwischen Tools	REST-API, Event Hub/Grid	REST-API, Data Connectors	REST-API	REST-API	REST-API
Wiederverwendbarkeit	kein Support	kein Support	Verlinkung über eindeutige ID, keine Typen möglich	Aspekte, Verknüpfungen zu Endpunkten	„Business Objects“ (MAS) + Verlinkung von Artefakten (ELM)
Betrieb: Cloud / on Premises	✓ / –	✓ / –	✓ / ✓ (als Komponente in Software kommerzieller Anbieter)	✓ / –	✓ / ✓ (beide)
Preis-/Lizenzmodell	Abrechnung pro Operation	Kategorien: Free, Basic, Standard; zudem „Tiered Bundles“ mit festem Monatspreis	– (Open Source)	nach Anzahl und Nutzung der digitalen Zwillinge	Lizenzierung über App-Points – abhängig von aktiven Nutzern

Alle Angaben beruhen auf Angaben der Hersteller gegenüber iX oder auf öffentlich zugänglichen Informationen.

Marktübersicht: IoT-Hubs

Produkt	Azure IoT Hub	IoT Greengrass	Eclipse Hono	Watson IoT Platform	xDT (Executable Digital Twin)
Hersteller	Microsoft	AWS	Eclipse	IBM	Siemens
unterstützte Kommunikationsprotokolle	MQTT, AMQP, HTTPS	MQTT, OPC-UA, Modbus-RTU (Custom)	MQTT, AMQT, HTTP, CoAP, Custom	MQTT, Custom	MQTT, OPC-UA
Betrieb: Cloud / on Premises	✓ / ✓	✓ / ✓ (als „IoT SiteWise Edge“)	✓ / ✓	✓ (bis Ende 2023) / ✓	k. A.
Lizenzmodell	Stufen: Basic und Standard (pro Einheit)	nach Geräteanzahl (0,18 US-Dollar/Gerät)	nein (Open Source)	keine neuen Abos mehr abschließbar	abhängig von SimCenter-Lizenz

Alle Angaben beruhen auf Angaben der Hersteller gegenüber iX oder auf öffentlich zugänglichen Informationen.

Tool die einzelnen Zwillinge jedoch File-basiert erstellt, ist dennoch eine Wiederverwendung durch Kopieren bestehender Dateien möglich. Dafür bietet Eclipse sogar eigene standardisierte Template-Files an. Die Angebote von Bosch und AWS erlauben zusätzlich die Wiederverwendung von Verknüpfungen zu Endpoints. Um einen Zwilling nun zu verwenden, müssen externe Services darauf zugreifen (siehe Dashboard, Planungsservice oder Predictive-Maintenance-Service in Abbildung 1). Jedes der vorgestellten Tools besitzt ein REST-Interface, das eine solche Interoperabilität zwischen dem Zwillingsinterface und externen Services ermöglicht.

Darüber hinaus ermöglicht das Azure-Interface über Event Hubs und Event Grids eine plattformsspezifische Umsetzung, um asynchron Daten aus den Zwillingen in weiteren Azure-Anwendungen zu verwenden. Da die Bosch-Software auf dem AAS-Standard aufbaut, bietet das Tool hier dieselben Funktionen wie Eclipse. Bei IBM geht der Interoperabilitätssupport hingegen nicht über die REST-Schnittstelle hinaus.

Oft ist es nötig, von einer Plattform auf eine andere zu wechseln. Je höher die Portabilität einer Plattform, desto einfacher ist dieser Umstieg. Bei den Produkten von Eclipse und Bosch ist es beispielsweise einfach möglich, in jede andere Plattform zu wechseln, die ebenfalls den AAS-Standard implementiert. Bosch definiert aber über die AAS hinaus noch Informationen im offenen Eclipse Semantic Modeling Framework Standard (ESMF),

weshalb die Zielplattform zur einfachen Portierung der Informationen ebenfalls diesen Standard implementieren muss. Beherrschen beide Tools die nötigen Standards, ist die Portierung einfach über Export der Zwillinge aus der Ausgangsplattform und anschließendem Import in die Zielplattform erledigt.

IoT-Hubs: Verknüpfung mit dem laufenden System

Die meisten Hersteller der Digital-Twin-Interfaces haben auch Produkte für die Verknüpfung des erstellten Zwillings mit laufenden Geräten im Angebot. Solche Tools sind auch als IoT-Hubs bekannt. Die Tabelle „Marktübersicht: IoT-Hubs“ gibt einen Überblick, welche Kommunikationsprotokolle die IoT-Hubs der einzelnen Hersteller einsetzen. Die Grundfunktionen von IoT-Hubs, namentlich die Datensynchronisation, das ferngesteuerte Ausführen von Operationen, die verschlüsselte Kommunikation sowie die Authentifizierung verknüpfter Geräte beherrschen alle berücksichtigten Produkte.

Hauptaufgabe der IoT-Hubs ist das Verknüpfen eines DT-Interface mit dem laufenden System, sie sind also Gateways zwischen physischen Geräten und der Cloud. Außerdem besitzen alle IoT-Hubs weitreichende Features zur Verschlüsselung dieser Kommunikation: Die Tools verschlüsseln die übertragenen Daten, die Verbindung zu den Geräten und verwalten die Authentifizierung verknüpfter Geräte. Zwillingsinterfaces benötigen zur Kommunikation mit Geräten

über den IoT-Hub also nur den eindeutigen Identifier des Geräts, während der IoT-Hub die Identifikation des konkreten Geräts auf Basis der IP-Adresse, die Auswahl des korrekten Kommunikationsstandards und die Absicherung der Kommunikation übernimmt. Das erfordert derzeit jedoch noch das händische Schreiben entsprechenden Codes auf dem Gerät. Dieser Code muss zum einen eingehende Operationsaufrufe korrekt verarbeiten und zum anderen vom Gerät angebotene Daten regelmäßig mit dem IoT-Hub synchronisieren.

Für die Kommunikation mit den Geräten unterstützen die Tools verschiedene standardisierte Verfahren. Während bei den anderen Features der Funktionsumfang der Produkte fast identisch ist, unterscheiden sich die Tools hier deutlich. Die meisten IoT-Hubs beherrschen die Standardprotokolle wie MQTT und HTTP(S), einige bieten zudem Support für AMQP oder die eigenständige Anbindung individueller Protokolle. Mit den weitverbreiteten Automatisierungsprotokollen wie MTConnect, OPC-UA oder Siemens S7 kommt jedoch kaum ein Hub zurecht. Nur Siemens xDT und AWS Greengrass bieten OPC-UA-Support.

Toolüberblick Verknüpfung mit Simulation

Neben dem laufenden System soll das Digital-Twin-Interface auch mit Simulationen der einzelnen Geräte verknüpfbar sein. Erst dadurch ergeben sich Anwendungen wie Predictive Maintenance und

Marktübersicht: Simulationstools

Produkt	SimCenter	RoboMaker	Simulink	3DEXperience	Gazebo
Hersteller	Siemens	AWS	Mathworks	Dassault	Open Robotics
Unterstützung 3-D-Modellierung	✓	✓	durch 3-D-Animation-Erweiterung	✓	✓
unterstützte Standards zur 3-D-Modellierung	✓ (k. A. welche)	STL, SDF, URDF, COLLADA	STL, FBX, DAE COLLADA, URDF, SDF	STL, FBX	STL, OBJ, COLLADA
Interaktionsschnittstelle	✓ (k. A. welche)	ROS	REST/MQTT	über Applications-Developer-Tooling	ROS/CLI
unterstützt FMI-Standard zur Interaktion	✓	–	✓	✓	über Open-Source-Projekt Gazebo-FMI
Installation: Cloud / on Premises	✓ / ✓ (manche Module cloudexklusiv)	✓ / –	✓ / ✓	✓ / ✓	– / ✓
Lizenzmodell	Lizenz pro Sitzung/Ausführung; Abrechnung monatlich/jährlich	nach Simulations-/GPU-Einheit	jährlich / Einmalzahlung nach Verwendungszweck (kommerziell oder akademisch)	dreistufiges Abo-Modell (Educational, Startups, Businesses)	– (Open Source)

Alle Angaben beruhen auf Angaben der Hersteller gegenüber iX oder auf öffentlich zugänglichen Informationen.

Optimierung, die die Verwendung digitaler Zwillinge richtig interessant machen. Für die technische Abbildung von Simulationen gibt es bereits etablierte Software wie Matlab Simulink, die 3DEXperience Suite von Dassault Systemes, oder die Open-Source-Alternative Gazebo. Durch den Hype um digitale Zwillinge gibt es inzwischen aber auch hier Konkurrenzprodukte großer Softwareanbieter, konkret das SimCenter von Siemens und den RoboMaker von AWS (siehe Tabelle „Marktübersicht: Simulationstools“).

Ein zentrales Feature der Simulationstools ist die dreidimensionale Repräsentation des simulierten physischen Objekts (3-D-Modellierung). Diese 3-D-Modellierung unterstützen alle untersuchten Tools. Bei Simulink ist dafür aber die Erweiterung Simulink 3D Animation notwendig. Alle Tools importieren und exportieren 3-D-Modelle in standardisierten Formaten, was den Modellaustausch mit anderen Tools der Kategorie ermöglicht.

Die wichtigste Funktion dieser Tools ist jedoch das Abbilden von Verhalten im Rahmen einer Simulation. SimCenter und 3DEXperience von Dassault als Sammlungen verschiedener Tools enthalten beide auch spezialisierte Simulationssoftware für den konkreten Anwendungszweck. Simulink und Gazebo haben dagegen ein integriertes Tool, das sich flexibel für verschiedene Einsatzszenarien eignet. Der Fokus von AWS RoboMaker liegt hingegen voll auf der Simulation von Robotern in industriellen Umgebungen.

Für den digitalen Zwilling ist interessant, wie das Digital-Twin-Interface mit den Simulationen programmatisch interagieren kann. Während Simulink hierfür lediglich eine klassische REST/MQTT-Schnittstelle anbietet, erlauben AWS und Gazebo Zugriff über eine Integration mit dem Robot Operating System (ROS). ROS implementiert ein eigenes Protokoll für

asynchrone Kommunikation, das Programme durch Packages für verschiedene Sprachen wie Python oder Java verwenden können. Beim SimCenter kann der digitale Zwilling über SDKs für verschiedene gängige Programmiersprachen auf die Simulationen zugreifen. Angaben darüber, welche Programmiersprachen man konkret unterstützt, macht Siemens aber nicht. Dassault erlaubt die Interaktion mit den Werkzeugen seiner 3DEXperience-Tools über ein selbst entwickeltes Application Developer Tool, stellt aber keine öffentlich beschriebene Schnittstelle nach außen zur Verfügung. Dassault unterstützt jedoch den Standard FMI, der Vorgaben an das angebotene Interface macht, um die Kommunikation mit Simulationstools zu vereinheitlichen. Auch Siemens, Simulink und Gazebo unterstützen diesen Standard, AWS aktuell noch nicht.

Fazit: Viele Lösungen machen das Leben schwer

Dieser Artikel zeigt, dass es viele verschiedene Tools gibt, die die einfachere Verwendung digitaler Zwillinge versprechen. Der Funktionsumfang unterscheidet sich, die Wahl der richtigen Tools hängt stark vom Einsatzzweck ab. Der Vergleich soll außerdem dabei helfen, zu verstehen, welche Tools für den sinnvollen Einsatz digitaler Zwillinge überhaupt notwendig sind.

Bei der Erstellung und Verwaltung digitaler Zwillinge über das DT-Interface sieht man bei den Tools einen Trade-off zwischen Unterstützung bei der Erstellung und Anbindung digitaler Zwillinge und der langfristigen Wartung der einmal erstellten Zwillinge. Außerdem gibt es hier Tendenzen zu großen Cloud-Anbietern wie Azure und AWS, da diese bereits ein Ökosystem bestehender Zusatztools anbieten, die sich so einfach an das

Zwillingen anbinden lassen. Der Nachteil dieser Anbieter ist jedoch, dass sie aufgrund ihrer plattformspezifischen Umsetzungen wenig Support bei der Übertragung von Zwillingen zu anderen Anbietern oder für die Anbindung externer Tools liefern.

Die Auswahl des IoT-Hubs, der die Anbindung des Zwillings an das laufende System ermöglicht, ist hingegen häufig von der Wahl des Zwillingsinterface bestimmt. Für ein Unternehmen, das beispielsweise den Azure Digital Twin Service verwendet, ist es am sinnvollsten, auch mit dem Azure IoT Hub zu arbeiten. Anbieter wie Eclipse und Bosch gestatten Anwendern etwas mehr Freiheit. Grundsätzlich bieten die untersuchten IoT-Hubs ähnliche Funktionen an. Die Entscheidung hängt hauptsächlich von den benötigten Schnittstellen der konkreten Hardware ab, die mit dem IoT-Hub kommunizieren soll.

Bei den Simulationstools liegt dabei die Entscheidung vor allem bei den unterstützten Austauschformaten, wobei hier die REST/MQTT-Schnittstellen von Simulink eine Integration mit bestehenden Digital-Twin-Interfaces am einfachsten gestalten. (jvo@ix.de)

Quellen

Daniel Lehner; Zweieieige Unterstützung; Digitale Zwillinge: Tool-Überblick und Anwendung in Azure Digital Twins; iX 5/2022, S. 140

DANIEL LEHNER

forscht als Doktorand an der Johannes Kepler Universität in Linz an den Potenzialen digitaler Zwillinge für cyber-physische Systeme.

