

# Einsatz eines Instandhaltungs–Cockpits am Beispiel einer Sondermaschine der Firma Xenon bei Continental Automotive Limbach-Oberfrohna

Jörg Stelzner<sup>1</sup>, Stefan Koppitz<sup>1</sup>, Colette Neubert<sup>2</sup>, Steffen Weiß<sup>2</sup>

<sup>1</sup>Xenon Automatisierungstechnik GmbH Dresden

<sup>2</sup>P ES LBO IT, Continental Automotive GmbH – Standort Limbach–Oberfrohna

## Zusammenfassung

Im Rahmen des Verbundforschungsvorhabens „Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme (S-CPS)“ ist in bilateraler Zusammenarbeit zwischen den Firmen Xenon und Continental Automotive GmbH - Standort Limbach-Oberfrohna ein Demonstrator für ein Ressourcen Cockpit der Instandhaltung entstanden, der im Folgenden näher erläutert wird.

Nach einer Übersicht zur systemtechnischen Darstellung des Demonstrator– Gesamtsystems werden die einzelnen systemtechnischen Komponenten und deren Funktionalitäten beschrieben. Besonderes Augenmerk wird dabei auf das Interface zur Selektion, Transformation und standardisierten Bereitstellung von Daten an der ausgewählten Pilotanlage gelegt. Die Einbindung des Demonstrator–Gesamtsystems in die Systemlandschaft am Continental Automotive GmbH - Standort Limbach-Oberfrohna wird beleuchtet sowie auch die externe Anbindung des Sondermaschinenbauers Xenon zur Fernwartung der ausgewählten Pilotanlage. Über die praktischen „use cases“ der Instandhaltung werden alle möglichen Aktivitäten und notwendigen Workflows zur Wartung und Instandsetzung der Pilotanlage erläutert. Diese Informationen werden dazu noch durch die Themen User, Rollenkonzepte und Berechtigungsvergabe angereichert.

In einem abschließenden Kapitel werden die erzielten Ergebnisse und die gewonnenen Erkenntnisse zusammengefasst und für die Nachnutzung aufbereitet.

## 1 Einleitung

Unsere Arbeiten zielten darauf ab, die für die Instandhaltung relevanten Datenströme des Produktionsequipments mit den relevanten Informationen für die an der Instandhaltung beteiligten Mitarbeiter zusammenzuführen, um daraus automatisiert und dynamisch Informationen und Entscheidungsvorlagen zu generieren, die eine effiziente Abarbeitung von Instandhaltungsvorfällen gewährleisten. Die Zusammenführung dieser Datenströme erfolgt im Back-

End in einem dafür adaptierten MES – System und im Front–End in einer mobilen Applikation, dem sogenannten Ressourcen – Cockpit. Über das Instandhaltungs-Cockpit sollte es auch möglich sein, alle für das Betreiben bzw. Instandhalten der Maschine erforderlichen Dokumentationen abzurufen und Handlungsanweisungen für die Beseitigung von Störungen anzuzeigen und zu ergänzen, um Wiederholstörungen effizient beheben und vorhandenes Wissen teilen und verwenden zu können.

Maschinendaten können je nach Interessenlage unter verschiedenen Gesichtspunkten ausgewertet werden. So sind z.B. für die Produktionssteuerung Stillstandzeiten und Verfügbarkeit mehr von Interesse als für die Instandhaltung, die auf Informationen über die Häufigkeit von Störungen angewiesen ist, um häufige Störungen und deren Ursachen zu beseitigen. Die neue Qualität der nun im MES-System vorhandenen Störmeldungen liegt darin, dass der Informationsgehalt der Störmeldung die komplette Ereigniskette im zeitlichen Ablauf umfasst und für Betreiber und Maschinen-Lieferanten im Fall der Störungsanalyse und Störungsbehebung permanent existente Informationen liefert.

Die Entwicklungsarbeiten für das System fanden dazu in verschiedenen Ebenen statt. Für die Bereitstellung der Maschinendaten wurde auf Maschinenebene von Xenon ein OPC-UA Server implementiert, der die Maschinendaten sammelt und über eine universelle Schnittstelle an einen OPC-UA Client übergibt. Auf der Betreiberseite wurde durch Continental ein OPC-UA Client implementiert, der die Daten über das lokale Netzwerk in einem dafür adaptierten MES-System ablegt. Alle Daten werden verschiedenen Endgeräten für die Auswertung zur Verfügung gestellt.

In den folgenden Abschnitten wird die Architektur und Funktionsweise des entwickelten Systems beschrieben.

## 2 Darstellung des Gesamtsystems

Die Einbindung des S-CPS in die bestehende Systemlandschaft der Continental Automotive GmbH - Standort Limbach-Oberfrohna ist, wie in Abbildung 1 dargestellt, zu beschreiben.

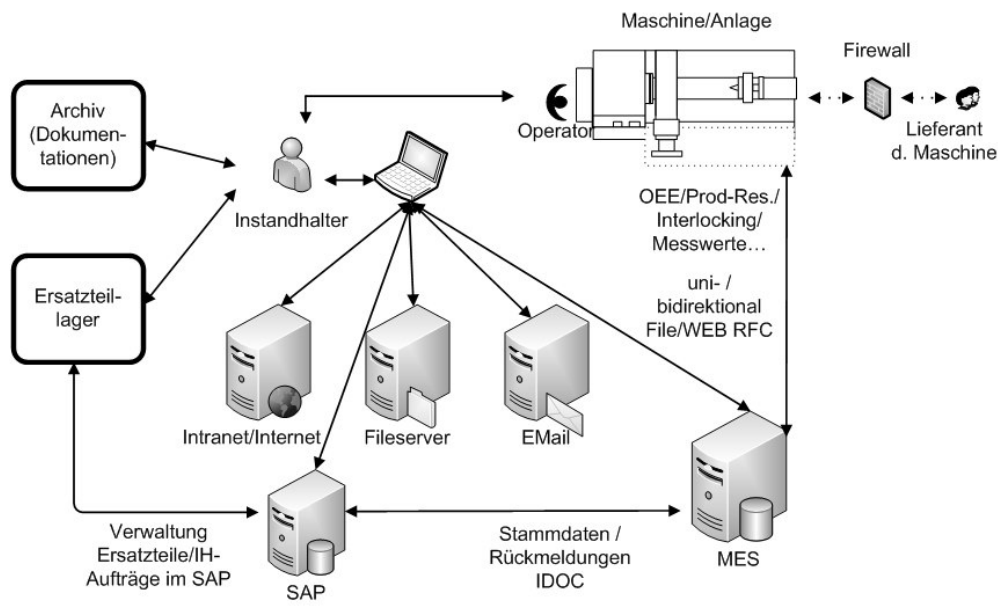


Abbildung 1: Übersicht der Systembestandteile - Basis zur Integration des S-CPS Systems (Quelle: Neubert S-CPS Lastenheft, 2014)

Der wichtigste Bestandteil des S-CPS Systems ist das Interface unserer Pilotanlage zum Continental - MES-System (Back-End). Die Pilotanlage des Herstellers Xenon wurde per Software in die Lage versetzt, neben produkt- und prozessbezogenen Informationen auch zustandsbezogene Informationen weiterzugeben. Diese Informationen werden mittels eines Gateways und dem Continental - Fertigungsclient an der Maschine visualisiert und in die MES-Datenbank übertragen. Der Bediener der Anlage entscheidet, ob die visualisierten Zustandsmeldungen der Anlage an die Kollegen des Instandhaltungsteams weitergegeben werden müssen. Dazu nutzt der Bediener eine Software-Oberfläche, die Teil des Front-Ends des Gesamtsystems ist. Wird das „Flag“ zur Weitergabe der Information an das Instandhaltungsteam gesetzt, wird diese dedizierte Meldung in der Datenbank „markiert“ und für den Kollegen der Instandhaltung sichtbar und bearbeitbar.

Das Instandhaltungsteam nutzt für die Weiterverarbeitung der zustandsbezogenen Informationen der Maschinen und Anlagen das dafür entwickelte Ressourcen-Cockpit (Front-End), sinnvollerweise als App auf den Tablets unserer Firma. Die Informationen, die in der Datenbank als relevant „markiert“ sind, werden auf dem Ressourcen-Cockpit als Liste und im Layout des Standortes

als Standortinfo der betroffenen Anlage visualisiert. Soweit hinterlegt, kann hier ein Handlungsleitfaden zur Behebung möglicher fehlerhafter Zustände hinterlegt sein, es ist aber ebenso möglich, auf alle Dokumente zur Maschinen- und Anlagendokumentation (Handbücher, E-Pläne, Projekte der Steuerungen ...) zuzugreifen.

Ist ein größerer Eingriff erforderlich, kann dies über das Anlegen eines Instandhaltungsauftrages per WEB Transaktion direkt im SAP (ERP-System) erfolgen, genauso wie die Fertigmeldung eines solchen Auftrages. Auch weitere Informationssysteme des Unternehmens sind über die App erreichbar (Email – System, Intranet ...).

Fallbezogen ist es notwendig, den Service des Maschinen- und Anlagenlieferanten in die Zustandsklärung einzubeziehen. Am Standort Limbach–Oberfrohna wurde dazu ein System der Fernwartung implementiert, das es erlaubt, kontrollierte gesicherte Verbindungen nach extern aufzubauen und damit einen Remote-Zugriff des Anlagenlieferanten zu ermöglichen. Diese Verbindungsfreigabe kann ebenso über die App durch den Instandhalter direkt vor Ort an der Maschine erfolgen.

Mit diesem Gesamtsystem ist es uns gelungen, alle notwendigen Tätigkeiten der Instandhaltung effizient und mobil zu gestalten, sodass die Verfügbarkeit von Informationen erhöht wird, zeitnah Reaktionen erfolgen können und Wegezeiten auf ein absolutes Minimum reduziert werden.

## 3 Back-End

Die Hard- und Software im Back-End, direkt an der Maschine, stellt ein Interface zur Selektion, Transformation und standardisierten Visualisierung und Weitergabe von Daten an der ausgewählten Pilotanlage zur Verfügung.

### 3.1 Datenbereitstellung auf Maschinenebene

Bei den Daten, die zwischen Maschine und MES-System ausgetauscht werden, handelt es sich sowohl um Stör- als auch um Betriebsmeldungen. Diese sollen im Weiteren als Alarme bezeichnet werden.

Alarme werden an der Maschine ausgelöst, wenn bestimmte Bedingungen erfüllt werden. Beispielhaft sei hier der Druckabfall an der Anlage genannt. Der OPC-UA-Server registriert diese Alarmmeldung von der SPS durch die Änderung einer Flanke und leitet dieses Signal an den Client weiter. Dabei kommuniziert der OPC-UA-Server mittels eines SIEMENS S7-Treibers mit der SPS. Der Alarm kann an der Anlage mittels Knopf auf dem Bedienpult quittiert

werden. Diese fallende Flanke wird wiederum vom OPC-UA-Server registriert und durch ein entsprechendes Signal an den Client weitergeleitet.

Ein Alarm besteht dann aus dem Zeitpunkt des Kommens/Auftretens, dem Zeitpunkt des Gehens/Quittierens, dem Alarmtext, dem Alarmort und dem Alarmlevel. Der Alarmlevel unterscheidet zwischen Stör- und Betriebsmeldungen. Störmeldungen führen unweigerlich zum Stillstand der Maschine bis das Problem behoben wurde. Eine Betriebsmeldung kommt einer Warnung gleich, die nicht sofort zum Stillstand führt. Denkbar wäre hier ein Materialmangel an der Maschine.

## 3.2 Datenaustausch zwischen Maschine und MES

### 3.2.1 OPC-UA – die Datenschnittstelle

Die OPC-UA (Open Platform Communications Unified Architecture) ist eine standardisierte Software-Schnittstelle für den Austausch von Daten und Informationen, insbesondere im industriellen Umfeld zur Automation, Steuerung und Kontrolle von Produktionsanlagen mit ihrer Umgebung und einzelner Anlagenbestandteile untereinander.

Namenhafte Firmen der Automatisierungstechnik haben sich 1996 zur OPC-Foundation zusammengeschlossen, um gemeinsam einen industriellen Standard zu schaffen, der einen sicheren, zuverlässigen, hersteller- und plattformunabhängigen Datenaustausch ermöglichen soll. Weiteres Ziel der Foundation ist es, die OPC-UA Standards weiter zu entwickeln und sie im industriellen Umfeld zu etablieren und auszubauen. Außerdem sollen zukünftig cloud-basierte Lösungen ermöglicht werden.

Der OPC-UA-Standard besteht aus mehreren Software-Schnittstellenspezifikationen wie Data Access (DA), Alarms & Conditions (AC), Data Exchange (DX) und weitere.

Durch OPC-UA als internationaler Industrie-Standard für die Datenkommunikation bringt dieser eine semantische Interoperabilität in die Cyber-Physical-Systems und trägt entscheidend zum Erfolg der 4. industriellen Revolution der Gegenwart bei.

Motivation und Grund für den Einsatz von OPC-UA im Pilotprojekt war vor allem die weite Verbreitung und breite Akzeptanz dieses Industrie-Standards. OPC-UA bietet eine einheitliche plattformunabhängige service-orientierte Architektur im Rahmen vom industriellen Internet der Dinge (IIoT). Mit OPC-UA wird ein optimiertes, robustes und fehlertolerantes Protokoll mit Sicherheitsmechanismen bereitgestellt. Es lässt sich eine einfache Kopplung zwischen

diversen Ereignisquellen realisieren. Die Datenübertragung ist durch die Verwendung von Eventmechanismen und ohne pollenden Zugriff schnell und performant. Zudem ist die Implementierung des OPC-UA-Kommunikationsstacks in .NET bereits vorhanden und erleichtert den Einstieg enorm. Informationen können mit objektorientierten Mitteln beschrieben werden. Der hier entwickelte OPC-UA-Server kann somit Grundlage für die standardisierte Schnittstelle zwischen Maschinenhersteller und MES sein.

Durch das ebenfalls bereits definierte Profil Alarms & Conditions im OPC-UA-Standard konnte der Austausch von Alarmen schnell vorangetrieben werden. Die sogenannten Companion Standards ermöglichen die Definition von herstellerspezifischen Profilen und somit eine gewünschte Erweiterung des Standards.

### 3.2.2 Die Pilotanlage als OPC – Server

Die Pilotanlage wird im Gesamtsystem als OPC-UA-Server (Abbildung 2) identifiziert. Die Software ist als .NET-Applikation (C#) implementiert, läuft auf einem Industrie-PC und ist per Ethernet mit dem Kunden- und dem Maschinennetzwerk verbunden.

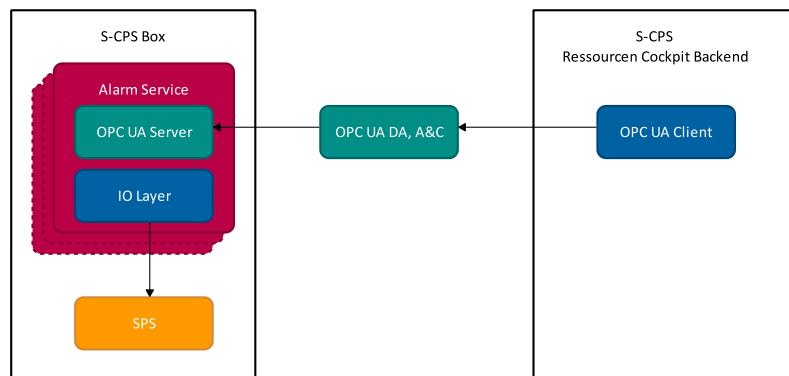


Abbildung 2: Schematischer Aufbau zum OPC-UA-Server

Die Maschine mit der SPS dient dabei als Datenquelle. Ein interner Client greift auf die Datenendpunkte (Tags) der SPS zu und leitet Änderungen an den Alarm Service weiter. Der Alarm Service als OPC-UA-Server meldet dem OPC-UA-Client per Events Alarme und stellt die Alarmdatenbank zum Abruf bereit. Das Daten- und Informationsmodell des OPC-UA-Servers wird von einer Konfigurationsdatei im XML-Format geladen.

In Abbildung 3 ist der Ablauf für die Interaktion mit dem System dargestellt. Dabei nutzt der OPC-UA-Client die Funktionen des OPC-UA-Servers. Beim **Push** (passiv) registriert sich der Client beim Server für ein bestimmtes Event. Der Server sendet dann bei Erfüllen der definierten Bedingung ein Event an den Client. Ein Polling durch den Client ist nicht erforderlich, da bei einer Änderung das Signal vom Server geliefert wird. Diese Funktion muss für den Empfang von kommenden und gehenden Alarmen verwendet werden. Beim **Pull** (aktiv) greift der Client auf Daten des Servers zu. Diese Funktion muss für den Zugriff auf die Alarmdatenbank verwendet werden.

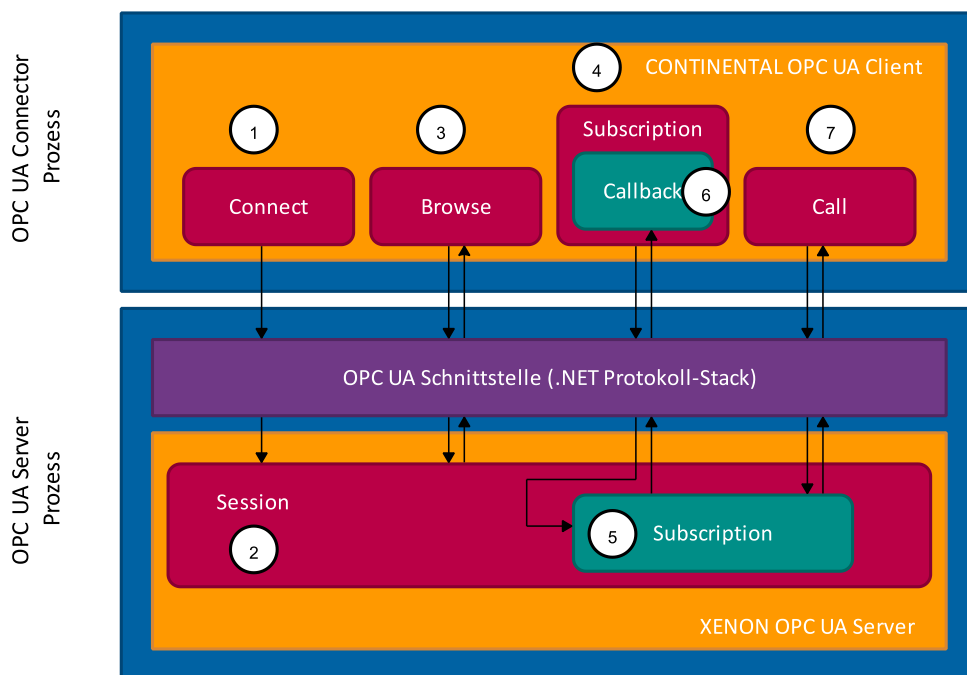


Abbildung 3: Struktur und Verhalten des Systems

Zur Umsetzung der Zielstellung musste der OPC-UA-Adressraum (Abbildung 4) erweitert werden. Da der ConditionType (def. in der OPC-UA-Spezifikation Teil 9) nicht mit dem angestrebten Alarmtyp in Einklang zu bringen war, wurde ein neuer Typ AlarmEventType zur Definition von Alarm Events eingeführt. Der Typ erbt vom Typ BaseEventType und übernimmt somit dessen Variablen, Methoden und Objekte.

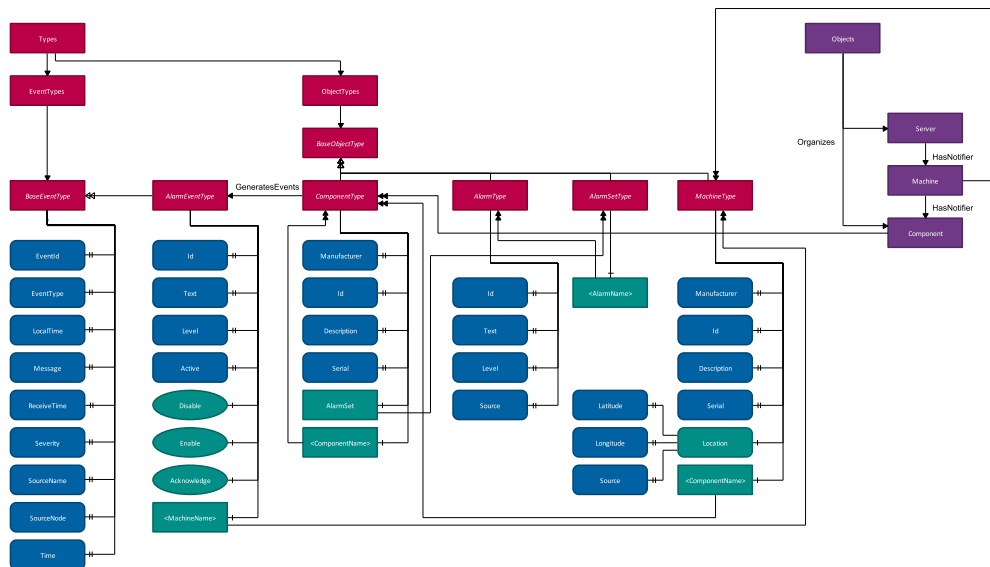


Abbildung 4: Adressraum des OPC-UA-Servers

Besondere Beachtung sollte das Objekt `<MachineName>` finden. Mit diesem wird der Ort des Auftretens kodiert. Das Objekt hat als Typdefinition den `MachineType`. Dieser Typ bildet die Struktur der Maschine/Anlage ab, da dieser rekursiv wieder aus `Components` bestehen kann. Ein `Component`-Objekt hat als Typdefinition den Typ `ComponentType`. Ein `Component` stellt die innere Struktur der Maschine/Anlage dar und kann somit z. B. ein Modul, eine Zelle oder eine Baugruppe sein. Ein `Component`-Objekt generiert dann auch die Events vom Typ `AlarmEventType`.

Für das Auslesen der Alarmdatenbank wurden die Objekttypen `AlarmType` und `AlarmSetType` eingeführt. Ein `AlarmSet` besteht dabei wiederum aus `Alarms`. Ein `Component` besitzt ein solches `AlarmSet` und stellt dieses dem Client zur Verfügung. Damit ist auch ein eindeutiger Bezug zum Ort des Auftretens gegeben.

Zur Abbildung der Maschinenstruktur im OPC-UA-Namensraum werden sogenannte `Areas` verwendet. Die Referenz `HasNotifier` wird verwendet, wenn das Ziel der Referenz ein Eventmeldeobjekt ist. Die Referenz `HasEventSource` wird verwendet, um auf die Quelle von Events zu zeigen. Die Quelle von Events können Variablen oder, wie hier, Objekte sein.



### 3.2.3 Das MES als OPC – Client

#### Allgemeines

Der OPC-UA-Client ist das Gegenstück zum OPC-UA-Server, der sich mit diesem verbindet, um Daten zwischen ihm und ein am Client angeschlossenes Back-End-System (z.B. ein MES-System) austauschen zu können. Die Verbindung erfolgt ausschließlich über das TCP/IP Netzwerkprotokoll. Der Client ist der aktive Partner der Datenkommunikation. Er verbindet sich mit dem Server und holt die für ihn bestimmten Daten ab oder übergibt Daten an den Server. Bei Verbindungsverlust wird versucht, die Verbindung automatisch wieder herzustellen. Der Client kann sich bei Bedarf mit mehreren Servern verbinden. In Abbildung 5 ist der generelle Aufbau skizziert.

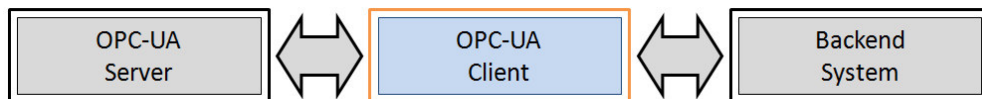


Abbildung 5: OPC-UA Client als Gateway

Im Pilotprojekt wurde eine Datenkommunikation zwischen dem OPC-UA-Server der Fertigungsanlage (Xenon) und dem MES-System der Continental Automotive GmbH als Gateway (OPC-UA-Client), wie in Abbildung 6 gezeigt, verwirklicht. Die Anbindung an das firmeneigene MES-System ist zwingend notwendig, weil dieses das zentrale Datenbanksystem der Fertigung am Standort Limbach-Oberfrohna ist und in diesem alle produktionsrelevanten Daten, so auch Maschinentalarme, zu speichern sind. Außerdem kann auf ein vorhandenes Portfolio an nützlichen Daten und Anwendungen (z.B.: OEE) zurückgegriffen werden.

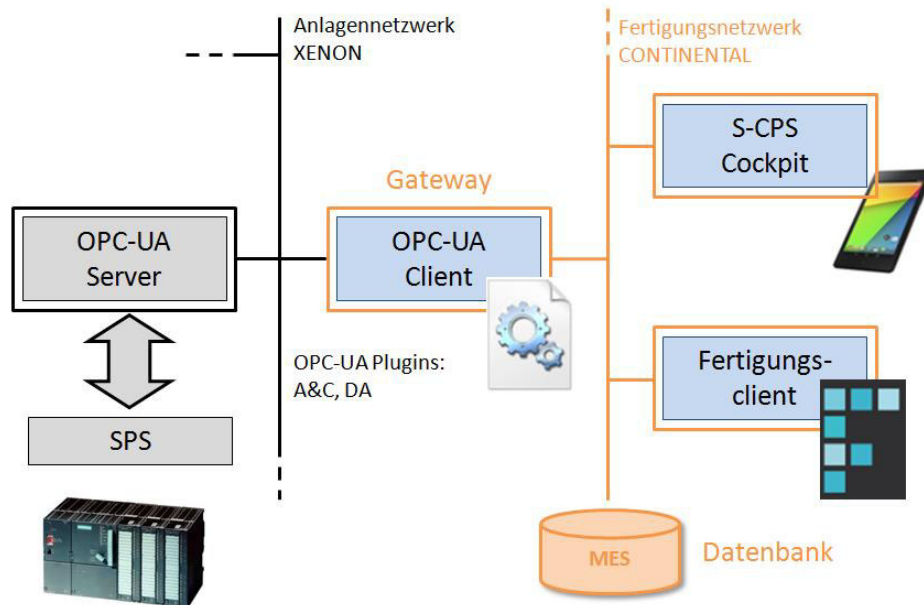


Abbildung 6: OPC-UA-Systemlandschaft

Mit dem Gateway werden ausschließlich Maschinenalarme der Pilotanlage an das MES-System übergeben. Für den spezifizierten Anwendungsfall wurden die OPC-UA-Plugins „Alarms&Conditions“ und „Data Access“ implementiert. Sämtliche Alarme der Anlage werden sofort über den OPC-UA-Client im MES-System gespeichert. Da nur Alarme, die nicht nach einer Erstinstandsetzungsfrist durch den Maschinenbediener behoben werden konnten an das S-CPS Cockpit weitergeleitet werden dürfen, wurde eine zusätzliche Anwendung für den Maschinenbediener geschaffen. Über diese Anwendung (ein firmeneigener Fertigungsclient) werden alle erfassten Alarme der betroffenen Anlage visualisiert und verwaltet. Können Alarme nicht durch die Erstinstandsetzung behoben werden, dann werden diese durch den Maschinenbediener über die Applikation an das S-CPS Cockpit der Instandhaltung weitergeleitet. Da alle drei Komponenten das gleiche MES-Datenbanksystem nutzen, erfolgte die Integration mit relativ wenig Aufwand.

### *Ablauf*

Detektiert die Anlage einen Alarm, so wird dieser im OPC-UA-Server registriert. Bei aktiver Verbindung des OPC-UA-Client (Gateway) holt sich dieser den Alarm im Status „KOMMEN“ ab und lastet ihn mit Datum/Uhrzeit, Ursache/Grund, Gewichtung, Typ und Klasse im MES-Datenbanksystem ein.

Um eine aufwendige Stammdatenpflege für Alarmer im MES-System zu minimieren, besteht die Möglichkeit, diese direkt von der Anlage zu übernehmen.

Der im MES-System gespeicherte Alarm wird am Fertigungsclient angezeigt. Wird der Alarm innerhalb der Erstinstandsetzungsfrist behoben, so ist der Alarm an der Anlage zu bestätigen (Standardbedienkonzept), was der OPC-UA-Server mit Status „GEHEN“ registriert. Ein erneutes Übernehmen dieses Alarms durch das Gateway führt dazu, dass der vorhandene Alarmstatus im MES-System zusätzlich mit Status „GEHEN“ versehen wird. Ein Refresh des Fertigungsclients wird daraufhin die visualisierte Alarmliste aktualisieren.

Nicht behobene Alarmer werden durch das Bedienpersonal über den Fertigungsclient an das S-CPS Cockpit weitergeleitet. Daraufhin wird das Fachpersonal der Instandhaltung über das S-CPS Cockpit informiert und kann seine Arbeit zur Behebung des Fehlers aufnehmen. Nach erfolgreicher Wartung ist der Alarm an der Anlage ebenfalls mit „GEHEN“ zu bestätigen, was auch zur Aktualisierung des MES-Datenbestandes und des Fertigungsclients führt.

### *Implementierung*

Das OPC-UA-Client-Gateway wurde als Windows-Service auf Basis .Net/C# und der von der OPC-Foundation verfügbaren OPC-UA-API (Kommunikationsstack) entwickelt und beinhaltet Programmlogiken zur Verbindung und zum Datenaustausch mit dem OPC-UA-Server, der MES-Datenbank und darüber hinaus Businesslogik für individuelle und anwendungsspezifische Erweiterungen inklusive Konfiguration und Setup. Der schematische Aufbau ist in Abbildung 7 dargestellt.

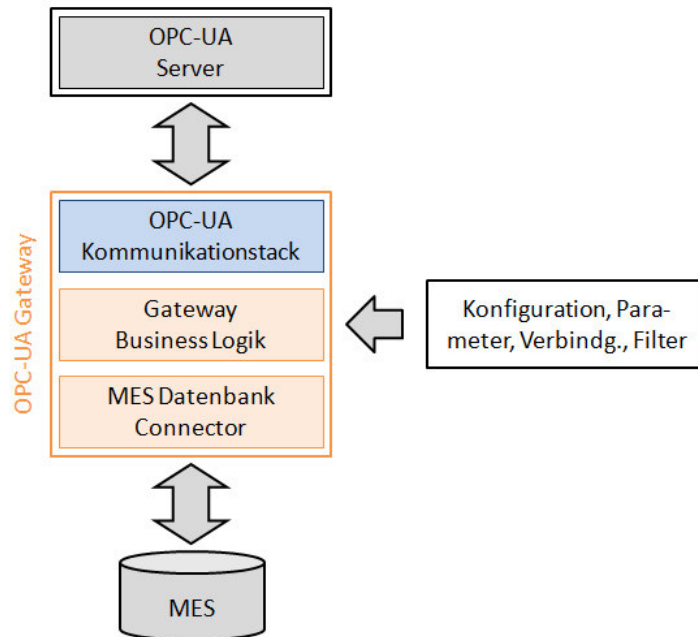


Abbildung 7: OPC-UA Client Gateway als Windows-Service

Außerdem ist das Gateway in der Lage, sich mit mehreren OPC-UA-Servern zu verbinden, so dass mit nur einem Gateway-Rechner mehrere Anlagen oder Anlagenteile abgedeckt werden könnten. Filtermöglichkeiten für Alarmquelle, Gewichtung, Klasse und Typ stehen ebenfalls zur Verfügung.

Für das Pilotprojekt „XENON Antriebsstrang Modul 33“ wurden OPC-UA-Server und -Client auf einem Rechner, der sich im Anlagennetzwerk befindet, installiert. Eine getrennte Installation auf zwei im Netzwerk befindlichen Rechnersystemen ist ebenfalls möglich.

## 4 Front-End

Das Front-End für die Interaktion mit dem Gesamtsystem besteht aus mehreren Komponenten. Hierzu gehört die Oberfläche für den Bediener der Anlage an der Maschine, die mobile Applikation (App) für das Team der Instandhaltung und die Sicht des Managements auf die Gesamtsituation.

## 4.1 Front-End Maschine

### 4.1.1 Der Continental Fertigungsclient

Am Fertigungsstandort Continental Automotive GmbH - Limbach-Oberfrohna wird ein firmeneigener, standardisierter Fertigungsclient (Abbildung 8) eingesetzt. Mit dessen Hilfe werden sämtliche produktions- und qualitätsrelevanten Daten der Produktion erfasst und in einem zentralen Datenbanksystem (MES-System) gespeichert, berichtet und archiviert.

Im Pilotprojekt dient er der Visualisierung aller an der Pilotanlage aufgelaufenen Störungen und der Weiterleitung an das Cockpit im Fall der Nichtbehebung innerhalb der Bedienererstinstandsetzungsfrist.

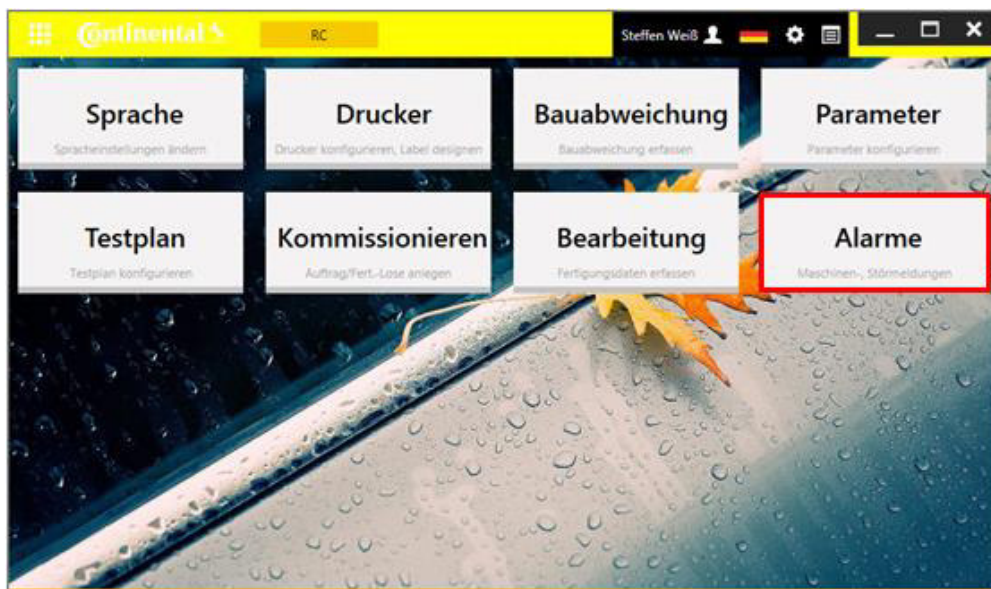


Abbildung 8: Continental Automotive GmbH Fertigungsclient

### 4.1.2 Das Alarm Applet

Für die Überwachung und Verwaltung von Maschinenalarmen durch das Maschinenbedienpersonal wurde eigens eine Applikation (Applet „Alarme“, Abbildung 9) für den Fertigungsclient entwickelt. In Abbildung 10 ist der Einsatz am BDE-System an der Pilotanlage zu sehen.

Dieses Applet hat über die MES- Datenbank direkten Zugriff auf alle aufgelaufenen Alarme und kann sich zusätzlich über WCF (Windows Communication Foundation) mit dem OPC-UA-Client-Gateway zu dessen Kontrolle und Steuerung verbinden.

The screenshot shows the 'ALARME' (Alarms) interface. The top bar includes the Continental logo and 'RC'. The user 'Steffen Weiß' is logged in. The main area is titled 'ALARME' and contains the following information:

**Informationen:**

- Gateway (Name): 2: OPC-UA ALARM PCR45 Antrieb / Kontaktierung
- Maschine/Anlage: IAK/S3/M33 (34208882-33)
- Status: Online
- Transfer (PDB): aktiv

**Alarme und Zustände:**

an FM	Zeit	Alarmcode	Text/Bezeichnung	Wichtigkeit	Klasse	Typ
	27.09.16 14:35:00	6066	St06-1 Spannzange nicht in Grundstellung (E306.0->0, E306.1->1)	Low	Störmeldung	AlarmCond
	27.09.16 14:30:00	7023	St07 Absaugung Saugstrom nicht gesperrt (E266.1->1)	Low	Störmeldung	AlarmCond
	27.09.16 14:25:00	7023	St07 Absaugung Saugstrom nicht gesperrt (E266.1->1)	Low	Störmeldung	AlarmCond
	27.09.16 13:55:00	7023	St07 Absaugung Saugstrom nicht gesperrt (E266.1->1)	Low	Störmeldung	AlarmCond
	27.09.16 13:50:00	7023	St07 Absaugung Saugstrom nicht gesperrt (E266.1->1)	Low	Störmeldung	AlarmCond
	27.09.16 13:49:00	7023	St07 Absaugung Saugstrom nicht gesperrt (E266.1->1)	Low	Störmeldung	AlarmCond
	27.09.16 12:51:00	11167	St11 Kamera nicht bereit	High	Störmeldung	AlarmCond
	27.09.16 12:45:00	11167	St11 Kamera nicht bereit	High	Störmeldung	AlarmCond
	27.09.16 11:37:00	11167	St11 Kamera nicht bereit	High	Störmeldung	AlarmCond
	27.09.16 11:35:00	5001	St05 Spritzschutz nicht in Servicestellung (E234.2->1)	MediumLow	Störmeldung	AlarmCond
	27.09.16 11:30:00	5001	St05 Spritzschutz nicht in Servicestellung (E234.2->1)	MediumLow	Störmeldung	AlarmCond

Below the table, there are filters for 'Zeitspanne' (8 h), 'Wichtung' (Min), 'Klasse' (all), and 'Typ' (all). At the bottom, there are function buttons: F1 Aktualisieren, F2 Service initialisieren, F3 Gateway starten, F4 Gateway beenden, F5 Transfer aktivieren, F6 Transfer sperren, F7 Übertragen, F8 Zurücksetzen, F9, and F10 Speichern.

Abbildung 9: Fertigungsclient - Alarmapplet

Die komplette Bedienung des Fertigungsclients ist durch Benutzerrollen abgesichert. Eine Benutzung durch nicht-autorisiertes Personal wird damit unterbunden. Das Applet kann sich mit mehreren gleichgearteten Diensten verbinden. Um das passende Gateway (zur Anlage zugeordnet) auszuwählen, bedient man sich der Steuerelemente im oberen Bildschirmbereich. Dort werden auch zusätzliche Informationen wie:

- verbundene Anlage
  - Betriebszustand des OPC-UA Gateway
  - Verbindungstatus zum MES-System und
  - Informationen zu den Datenqueues
- angezeigt.

Im mittleren Fensterbereich werden in der linken Liste alle Alarmer, die noch nicht geschlossen wurden (fehlender Status: GEHEN) zeitlich geordnet aufgelistet. In der rechten Liste werden alle Alarmer, summiert nach Störgrund, in einer kleinen Statistik dargestellt. In dieser werden auch Alarmer mit Status GEHEN angezeigt.

Die Behandlung von Anlagenstörungen bei der Continental Automotive GmbH am Standort Limbach-Oberfrohna ist so festgelegt, dass alle Störungen innerhalb einer Bediener-Erstinstandsetzungsfrist zu lösen sind. Erst wenn die Störung nicht behoben werden kann, wird die Fachabteilung Instandhaltung informiert, damit diese der Behebung der Störung nachgehen kann.

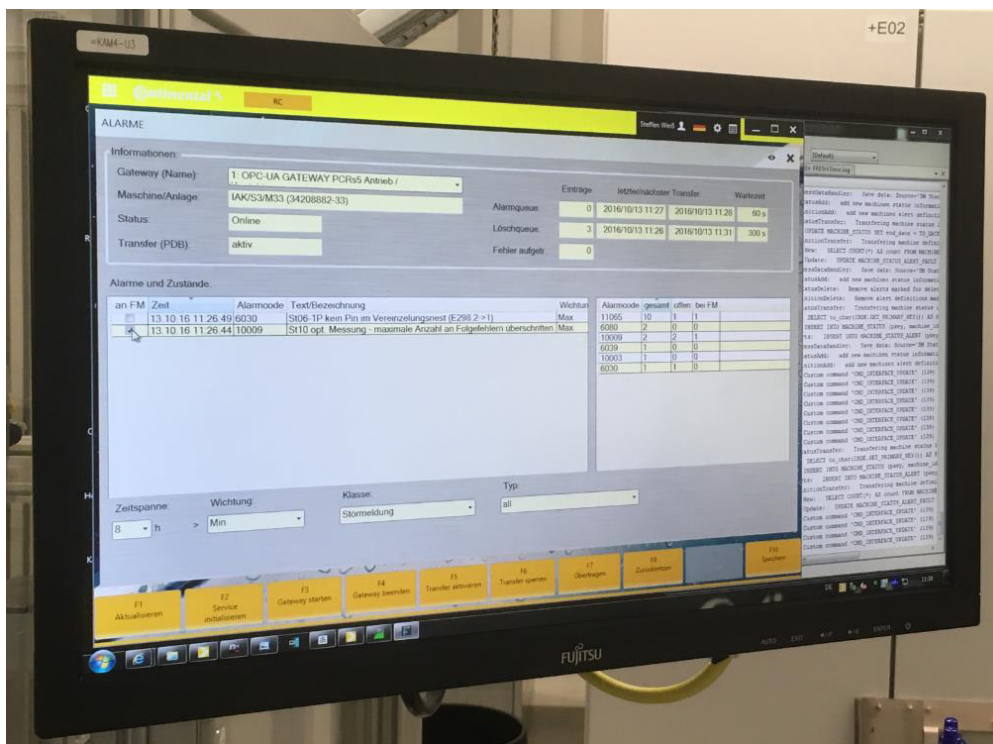


Abbildung 10: Fertigungsclient an der Xenon Pilotanlage

Dazu muss der Anlagenbediener über die Anwendung „Alarmer“ den betreffenden Alarm auswählen und über die Option „An FM“<sup>2</sup> markieren (s. Abbildung 11). Nach Speichern der Daten werden alle markierten Alarmer an das

<sup>2</sup> FM ... Facility Management (Abteilungsbezeichnung für die Instandhaltung)

Cockpit der Instandhaltung übergeben. Ein nachträgliches Ändern von an FM übergebenen Störmeldungen ist unterbunden.

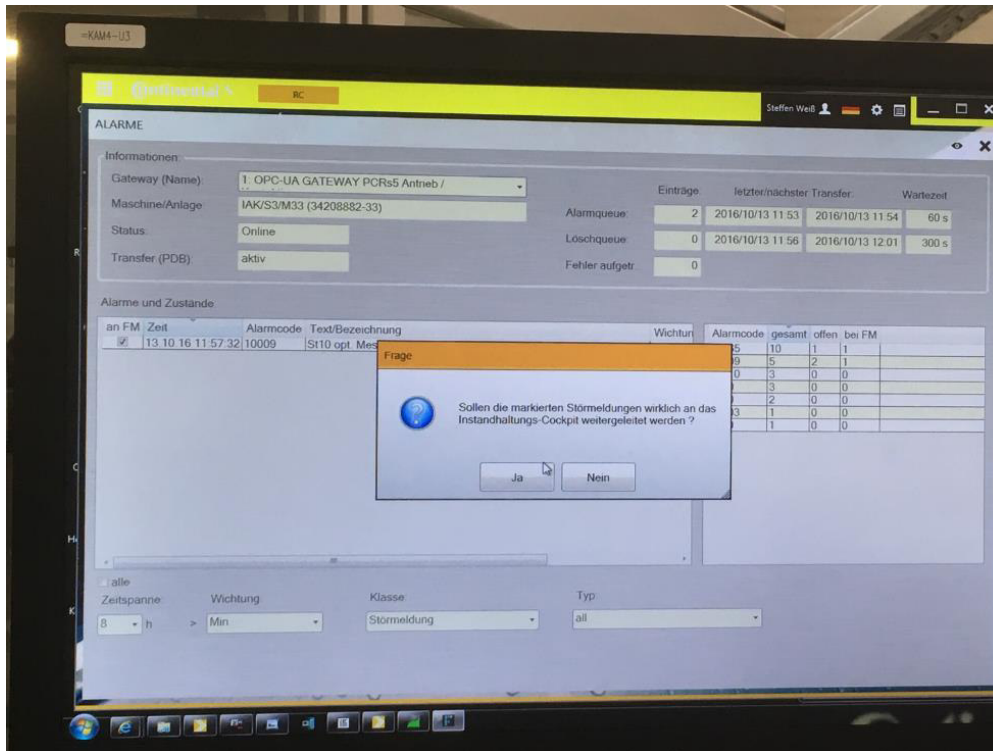


Abbildung 11: Störmeldung an das S-CPS Cockpit melden

Anschließend wird die Statistik neu berechnet und alle Ausgaben werden aktualisiert.

## 4.2 Front-End Instandhaltung/Management

### 4.2.1 Nutzung des S-CPS Ressourcen-Cockpits

Das Continental S-CPS-Ressourcen-Cockpit wurde am Standort Limbach-Oberfrohnna entwickelt. Die Entwicklung unter .Net/C# nutzt die zentral gehostete Oracle Datenbank unseres MES-Systems als Datenquelle und -senke. Die Kommunikation erfolgt über das lokale Netzwerk des Standortes und ist als bilaterale Kommunikation ausgelegt. Damit ist das Ressourcen-Cockpit als verteilte WEB-Applikation auf beliebig vielen Geräten in unserem Firmennetz



mit einem zentralen Informationsstand nutzbar. Dies ermöglicht uns die Nutzung der App auf Tablets für das mobile Team der Instandhaltung, genauso wie die Nutzung als Darstellung der Gesamtsicht für das Management sowohl mobil als auch am normalen Desktop-Arbeitsplatz. Das Layout orientiert sich an den Untersuchungen und Vorgaben, die im Rahmen des Verbundforschungsvorhabens erarbeitet wurden.

#### 4.2.2 Funktionen des S-CPS Ressourcen – Cockpits

Eine wesentliche Funktionalität des Ressourcen-Cockpits besteht in der Verarbeitung zustandsbezogener Informationen (Alarmer) von Maschinen und Anlagen. Über das standardisierte Back-End werden diese Informationen von den Maschinen und Anlagen erfasst und für die weitere Verarbeitung zur Verfügung gestellt. Der Bediener der Anlage entscheidet, ob diese Informationen an der Maschine verbleiben (Zustände werden durch den Bediener abgearbeitet) oder ob sie in die nächste Ebene der Abarbeitung gehoben werden müssen. Im letzten Fall werden diese Alarmer im Ressourcen-Cockpit gelistet (s. Abbildung 12) und können durch den Instandhalter in Bearbeitung genommen werden. Die Details und mögliche Handlungsleitfäden zur Behebung der Alarmzustände können über das Ressourcen-Cockpit abgerufen werden.

The screenshot displays the SCPS Cockpit interface. At the top, there is a header with the Continental logo and the title "SCPS Cockpit". Below the header, a navigation bar shows "5 Fehlermeldungen" (5 Error Reports) and several menu items: Übersicht, Alarme, Details, Dokumente, SAP Meldung, SAP Auftrag, Intranet, and Fernwartung. The main content area shows a table of error reports for machine 34208882-33. The table has columns for MACHINE, START\_DATE, END\_DATE, REMARK, FAULT, and SEVERITY. The data rows are as follows:

MACHINE	START_DATE	END_DATE	REMARK	FAULT	SEVERITY
34208882-33	13.10.2016 11:17:42			10009	1000
34208882-33	27.09.2016 11:20:00			1080	100
34208882-33	13.10.2016 10:46:47			11065	1000
34208882-33	14.10.2016 08:54:08			6045	1000
34208882-33	19.10.2016 17:28:00			6066	100

The interface also includes a "TEST!!!!" message, a "Refresh" button, and a footer with the text "Copyright©Continental Automotive GmbH - P ES INJ LBO".

Abbildung 12: Zu bearbeitende Alarmer auf Basis MES-Datenbank

Diese Funktionalität wird ergänzt durch weitere Funktionen, die wesentlich zur Effizienzsteigerung der Instandhaltungstätigkeiten führen. Zu diesen Funktionen gehört der mobile Zugriff auf die Maschinendokumentationen aller Maschinen und Anlagen. Dazu kann in einem Struktur-Baum über Hallen, Maschinenbezeichner und Dokumentart selektiert werden. Das gewählte Dokument wird je nach Berechtigung/Rolle nur angezeigt oder kann in einem Editor auch bearbeitet werden (Abbildung 13).

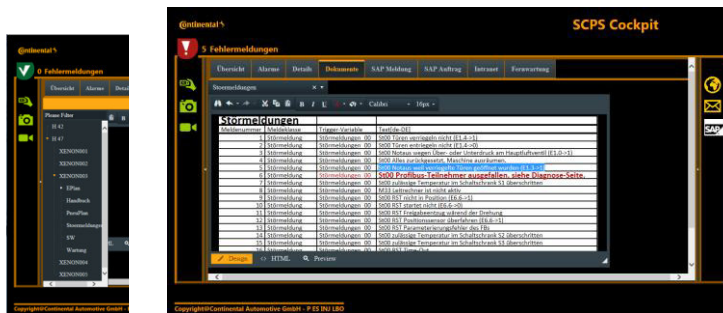


Abbildung 13: Selektion und Bearbeitung von Dokumenten

Da es fallabhängig immer wieder vorkommt, dass für die Bearbeitung von Alarmzuständen die Unterstützung des Maschinen-/Anlagenherstellers notwendig ist, ist am Standort Continental Automotive GmbH - Limbach-Oberfrohna ein System zur sicheren Remoteverbindung für diesen Anwendungsfall entwickelt worden. Diese Verbindungen müssen durch autorisierte Personen dediziert freigeschaltet werden. Um diese Freischaltung des Verbindungsaufbaus auch mobil verfügbar zu machen, ist auch diese Funktion in das Ressourcen-Cockpit integriert worden (Abbildung 14).



Abbildung 14: Dedizierte Freischaltung der Fernwartungszugänge

Als weitere Funktion ist die Abwicklung von Instandhaltungsaufträgen im ERP System SAP hinterlegt. Dazu kann direkt aus dem Ressourcen-Cockpit heraus eine Instandhaltungsmeldung (Auftrag) angelegt werden und dieser kann auch nach Abschluss der Arbeiten zurückgemeldet werden – damit ist im ERP System die Abarbeitung des Auftrages und dessen Inhalt hinterlegt und kann monetär bewertet werden. Diese Funktionen sind in der Abbildung 15 dargestellt.

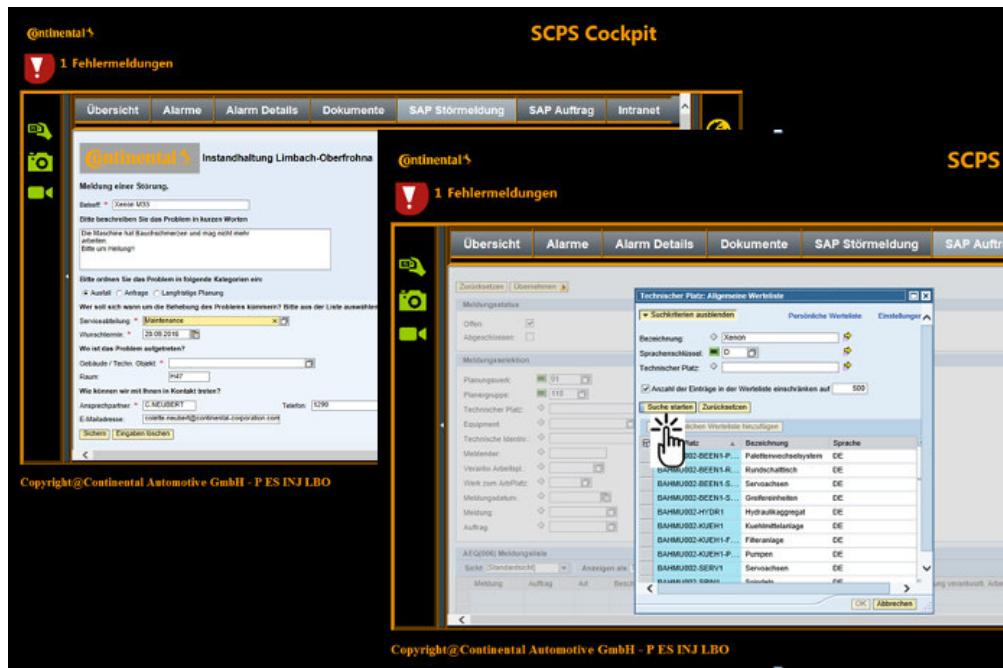


Abbildung 15: SAP Auftragsbearbeitung direkt aus dem Ressourcen-Cockpit

Auch auf weitere firmeneigene Systeme kann zugegriffen werden. Dazu gehören das Intranet, das Email System und andere.

Für das Management ist auf Basis des Systems ein Überblick über die existierenden Störungen und Alarmzustände in allen Fertigungsbereichen möglich. Ressourcen-Engpässe können erkannt und zeitnah behoben werden.

## 5 Zusammenfassung und Ausblick

Die Ergebnisse der Forschungstätigkeiten zeigen pilothaft die Möglichkeiten eines S-CPS-Systems für die Instandhaltung. Im Ergebnis wurden Grundlagen geschaffen, Metadaten einer Maschine für verschiedene Nutzergruppen bereitzustellen und anwendbar zu machen. Der entwickelte OPC-UA-Server kann je nach den speziellen Erfordernissen sämtliche Maschinendaten aus der Maschinensteuerung (SPS) und deren untergeordneten Systemsteuerungen bereitstellen und im zeitlichen Kontext an ein übergeordnetes System übergeben. Dabei ist es unerheblich, wie viele Module eine Maschine umfasst, da über den OPC-UA-Server die Daten mehrerer Module verwaltet und organisiert werden können. Es wurde im Rahmen dieses Projektes gezeigt,

wie die innerbetriebliche Bereitstellung der Daten für verschiedene Nutzergruppen als Unterstützung für die Produktion und den Instandhaltungsprozess erfolgen kann. Dabei stand neben der Bereitstellung der Daten vor allen auch die automatische Weiterleitung der Störungsmeldungen an das Instandhaltungs-Cockpit im Focus, um eine unverzügliche Störungsbehebung durch das Instandhaltungspersonal einleiten zu können. Bei Einsatz der entwickelten Systeme können in Zukunft Produktionsstillstandzeiten wesentlich reduziert werden. Der erreichte Entwicklungsstand des Demonstrators erlaubt es, dass in nachfolgenden Schritten die Integration des OPC-UA-Servers in alle anderen Module der Xenon Maschine erfolgen kann. Bisher betrifft dies jedoch nur die Maschinendaten, die durch ein von Xenon entwickeltes SPS Programm bereitgestellt werden.

Der entwickelte OPC-UA-Server stellt die Grundlage für den Standard der neuen Leitrechnergeneration (übergeordnetes System) bei Xenon dar. Der OPC-UA-Server kann universell zur Bereitstellung von Maschinendaten in allen anderen von Xenon gelieferten Maschinen eingesetzt werden und durch Modifizierung der Schnittstellen zu übergeordneten Systemen angepasst werden. Ein weiterer Entwicklungsschritt wird die Anpassung und Integration des OPC-UA-Servers in Xenon – fremde Maschinen und Module sein, um die Datenbereitstellung für komplette Fertigungslinien zu realisieren. Das kann nur in Zusammenarbeit mit den jeweiligen Systemherstellern erfolgen, da hierzu die Bereitstellung der Daten aus den jeweiligen Maschinen- und Systemsteuerungen erforderlich sind.

Der flächendeckende Einsatz dieses Systems für den kompletten Maschinen-/Anlagenbestand ist der nächste Schritt bei der Verwertung dieser Ergebnisse am Standort Continental Automotive GmbH – Limbach-Oberfrohna, wobei hier auch an die Erfassung weiterer Informationen im Sinne des „Condition Monitoring“ zu denken ist. Ist dieser Schritt abgeschlossen, geht es daran, eine weitere Integration dieses Systems in eine Service- und Systemlandschaft zu entwickeln und voranzutreiben.

## 6 Literaturverzeichnis

Neubert, C. (2014). S-CPS Lastenheft 2014 (Lastenheft für das Forschungsprojekt S-CPS). Limbach-Oberfrohna

## Autoren



### Neubert, Colette

Colette Neubert studierte an der TU Chemnitz Gerätetechnik und spezialisierte sich auf dem Gebiet der Lasertechnik und Informatik im Kontext medizintechnischer Gerätetechnik. Über die Themen Simulation, Software und Administration von Systemlandschaften im produzierenden Gewerbe erarbeitete Sie sich einen Gesamtblick auf die IT Landschaft produzierender Unternehmen und ist jetzt als IT Leiterin bei der Continental Automotive GmbH am Standort Limbach-Oberfrohna tätig.



### Stelzner, Jörg

Jörg Stelzner studierte Maschinenbau Fachrichtung Konstruktionstechnik an der Technischen Universität Chemnitz. Er arbeitet seit über 20 Jahren im Bereich Service, darunter über 15 Jahre in der Instandhaltung für Werkzeugmaschinen und Produktionsanlagen. Er leitet das Service-Center der Xenon Automatisierungstechnik GmbH in Dresden mit dem Schwerpunkt der Optimierung von Serviceprozessen, unter Verwendung moderner I&K Technologien.



### Weiß, Steffen

Steffen Weiß studierte Elektrotechnik an der TU Dresden und spezialisierte sich auf dem Gebiet der Informationstechnik. Nach dem Studium arbeitete er mehrere Jahre in der Softwareentwicklung für Firmen der Messtechnik, Forschung und Automobilindustrie. Seit mehr als 15 Jahren ist er bei Continental Automotive GmbH Limbach-Oberfrohna Bereich IT beschäftigt. Seine Aufgabe ist es, in MES Themen zu beraten und Applikationen zur Unterstützung der Produktion zu entwickeln.



### Koppitz, Stefan

Stefan Koppitz studierte Medien-Informatik an der Technischen Universität Dresden und schloß dieses Studium 2009 ab. Aktuell ist er bei der Firma Xenon Automatisierungstechnik GmbH tätig und konzentriert sich auf die Programmierung von SCADA, Linien- bzw. Leitrechnersystemen und MES-Funktionalitäten.