

# Flexible Gatewayschnittstelle zur Anbindung unterschiedlicher Netzwerke

Alexander Faul, Michael Weyrich  
IAS – Institut für Automatisierungstechnik und Softwaresysteme  
Universität Stuttgart  
Pfaffenwaldring 47  
70550 Stuttgart  
Tel.: +49 (0)711-685 67301  
Fax: +49 (0)711-685 67302

E-Mail: alexander.faul@ias.uni-stuttgart.de, michael.weyrich@ias.uni-stuttgart.de

**Abstract:** Im Beitrag wird das Konzept einer flexiblen Gatewayschnittstelle vorgestellt. Dieses Konzept ermöglicht es, ein Gateway mit unterschiedlichen Bussystemen zu verbinden, ohne dass hierzu die Systemsoftware des Gateways angepasst werden muss. Diese Flexibilität wird dadurch erreicht, dass die Schnittstelle und die notwendigen Nachrichten in einer Ontologie beschrieben werden, welche dann maschinell ausgewertet wird. Hierdurch können auch komplexere Funktionalitäten realisiert werden, wozu das Versenden mehrerer Nachrichten notwendig ist. Validiert wurde das Konzept anhand eines Gastronomie-Kaffeefullautomaten mit einem integrierten CAN-Bus. Hierbei konnten nicht nur einfache Funktionalitäten angesprochen werden, sondern auch komplexe Abläufe aus verschiedenen Nachrichten und Parametern realisiert werden sowie neue Funktionalitäten zur Laufzeit hinzugefügt werden.

**Stichworte:** Gateway, flexible Schnittstelle, Ontologie, Bussysteme, Interpreter

## 1 Motivation und Zielsetzung

Im Bereich von „Internet of Things“ (IoT) und „Industrie 4.0“ werden häufig sogenannte „Cyber Physical Systems“ (CPSs) verwendet. Diese unterscheiden sich von einem klassischen automatisierten System vor allem in ihrer Kommunikationsfähigkeit zu anderen Teilen des Gesamtsystems. Dies bedeutet, vor allem beim IoT, dass die CPSs eine Anbindung an das Internet besitzen. Für den Fall, dass automatisierte Systeme nicht über eine entsprechende Schnittstelle verfügen, sondern lediglich ein lokales Netzwerk / Bussystem besitzen, wird zum Beispiel in der IIRA (Industrial Internet Reference Architecture) [IIC2015] und der ETSI-M2M (European Telecommunications Standards Institut Machine-to-Machine) Architecture [ETSI2013] eine Gateway-Lösung vorgeschlagen. Hierbei stellt, wie aus Abbildung 1 ersichtlich, das Gateway die Verbindung zwischen dem lokalen Netzwerk bzw. Bussystem und dem übergreifenden Netzwerk, zum Beispiel dem Internet her.

Ein Gateway kann entweder individuell für den entsprechenden Anwendungsfall oder mittels eines entsprechenden Frameworks entwickelt und implementiert werden. Unabhängig vom Vorgehen

bei der Realisierung bleibt die Herausforderung, dass eine Software-Anwendung (App) eine Schnittstelle zu dem betroffenen automatisierten System benötigt. Hierbei liegt die Schwierigkeit nicht nur in den unterschiedlichen physikalischen Randbedingungen, zum Beispiel verschiedene Spannungspegel und Anzahl der Leiter, die die Schnittstelle des automatisierten Systems besitzt. Sondern vielmehr darin, dass selbst bei gleicher physikalischer Übertragung, zum Beispiel bei Verwendung des gleichen Bussystems, die ausgetauschten Nachrichten unterschiedliche Bedeutungen besitzen können. Bereits in [VDB2013] wird gezeigt, dass diese unterschiedlichen funktionalen Umfänge in den Geräten eine Vereinheitlichung erschweren. Am IAS wurde das im Folgenden vorgestellte Konzept entwickelt und prototypisch validiert. Ziel dieses Konzeptes ist es, eine einheitliche Anbindung unterschiedlicher Netzprotokolle und damit automatisierter Systeme an Kommunikationsnetzwerke zu ermöglichen.

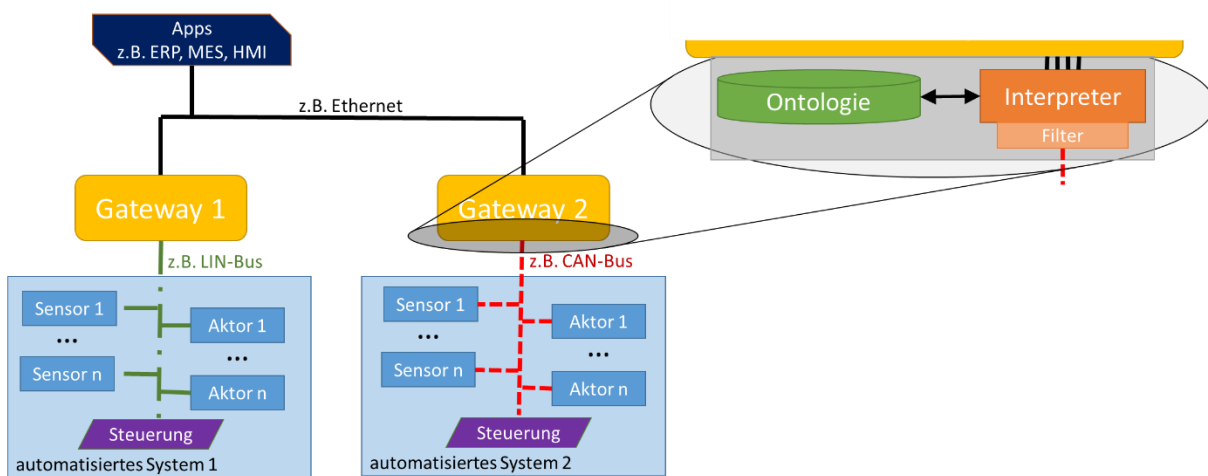


Abbildung 1: Grober Aufbau der Schnittstelle und deren Einsatzstelle

## 2 Stand der Technik

In diesem Kapitel werden verschiedene bestehende Ansätze für Gatewayschnittstellen vorgestellt.

Schon die OSGi Allianz definiert in der RFC 0196 [RFC0196] eine Abstraktionsschicht, welche einen einheitlichen Zugriff auf verschiedene Protokolle ermöglicht. Hierbei wird vorausgesetzt, dass für jedes Protokoll ein entsprechender Adapter bei der Abstraktionsschicht registriert wird. Diese Adapter übersetzen in das entsprechende Protokoll, vernachlässigen aber spezifische Eigenschaften der angebotenen Systeme.

Im Artikel „Schnittstellen flexibel realisieren mit FPGA“ [Bräut2011] stellt Bräutigam einen Ansatz vor, die Kommunikation zwischen einem Gateway und dem angeschlossenen Kommunikationsprotokoll mittels eines FPGAs und austauschbaren IP-Cores zu flexibilisieren. Hierbei liegt das Augenmerk vor allem darauf die Hardware flexibel einzusetzen. Die Software, in Form eines IP-Cores, muss hierbei weiterhin individuell entwickelt werden.

Riedl et al beschreiben in [RMZ2014] Anforderungen an die Verknüpfung von CPS als neuartige Architektur von automatisierten Systemen. Hierbei stellt die Anbindung der Hardware einen

wichtigen Punkt dar, wobei die Kombination verschiedener Middlewares als eine mögliche Lösung vorgestellt wird.

Die Plattform Industrie 4.0 verfolgt zusammen mit andern Fachgremien das Ziel eine einheitliche Sprache für die Kommunikation automatisierter Systeme und deren Komponenten zu definieren [Plat2015]. Ein erster Schritt hierzu ist die Verwaltungsschale [VDI2015], welche eine einheitliche Verwaltung von Industrie 4.0-Komponenten ermöglichen soll. Das hier vorgestellte Konzept könnte einen Teil dieser Verwaltungsschale darstellen.

### **3 Konzept**

Die Idee dieses Konzeptes ist es, eine maschinenauswertbare Beschreibung der Schnittstelle zwischen Gateway und automatisiertem System zur Interpretation von Nachrichten zu verwenden. Die Beschreibung setzt sich hierfür zum einen aus der Spezifikation der Schnittstelle selbst und den möglichen Funktionen des automatisierten Systems zusammen. Diese Beschreibung kann als Menge an Objekten mit Beziehungen dargestellt werden, so dass eine Ontologie eine sinnvolle Möglichkeit zum Speichern der Beschreibung darstellt. Zusätzlich zur Ontologie wird noch ein Interpreter benötigt, welcher die Nachrichten verarbeitet und hierzu mit der Ontologie interagiert. Damit der Interpreter Nachrichten an das Gateway von Nachrichten an andere Busteilnehmer unterscheiden kann, ist ihm ein Filter vorgeschaltet, dessen Parameter ebenfalls in der Ontologie definiert sind. Diese Aufteilung ermöglicht es, denselben Interpreter und Filter für unterschiedliche Protokolle zu verwenden, da die Unterschiede der Netzprotokolle durch die Beschreibung in der Ontologie abgebildet werden. Abbildung 1 zeigt den groben Aufbau und die Einsatzstelle der Schnittstelle. Im Folgenden werden die vorgestellten Komponenten detaillierter beschrieben.

#### **3.1 Ontologie zur Schnittstellenbeschreibung**

Mithilfe der Ontologie wird eine maschinenauswertbare Beschreibung der Schnittstelle in Form eines Graphen gespeichert. Eine Ontologie besteht aus zwei aufeinander aufbauenden Teilen. Der eine Teil besteht aus den Klassen der Ontologie, diese bilden eine allgemeine Beschreibung der Struktur und der Beziehungen. Der andere Teil besteht aus den Instanzen dieser Klassen, welche für die im ersten Teil definierten Eigenschaften explizite Werte besitzen. Dies ist vergleichbar mit der Beziehung zwischen Klassen und Objekten der objektorientierten Programmierung.

Zu der Beschreibung der Schnittstelle gehört das Nachrichtenformat, genauso wie die möglichen Operationen, welche das automatisierte System durchführen kann und die dafür notwendigen Nachrichten. Besitzen diese Operationen Parameter, gehören diese und Anweisungen zur Integration dieser in die Nachrichten ebenfalls zur Beschreibung. Diese Form der Speicherung bietet ein hohes Maß an Freiheitsgraden für den Aufbau des Graphen. Diese Freiheit erschwert aber die Auswertung des Graphen, da keine Kenntnisse über den Startpunkt und die grundlegende Struktur vorliegen. Es wird daher vom Konzept eine Grundstruktur vorgegeben, welche für die Beschreibung verfeinert und erweitert wird. Diese Grundstruktur bietet einen Startpunkt, von

welchem aus der Graph maschinell ausgewertet werden kann. Auf diese Weise wird es möglich, unterschiedliche automatisierte Systeme einheitlich anzusprechen.

In Abbildung 2 ist diese Grundstruktur abgebildet, welche mit einem allgemeinen Interface-Objekt beginnt. Dieses Objekt kann Nachrichten eines bestimmten Formates senden und empfangen. Mit einer Schnittstelle können im Allgemeinen Funktionalitäten des angeschlossenen automatisierten Systems angesprochen werden. Diese Funktionalitäten werden in der Ontologie als Operationen bezeichnet, welche Parameter besitzen können. Jede dieser Operationen hat eine Beziehung zu einer Nachrichtensequenz, welche wiederum aus einzelnen Nachrichten besteht. Dadurch ist es möglich komplexere Operationen, welche mehrere Nachrichten benötigen, zu realisieren. Die Parameter der Operationen haben wiederum eine Beziehung zu einer Transformationsanweisung, welche festlegt wie diese Parameter in die Nachrichten der entsprechenden Nachrichtensequenz integriert werden müssen. Zudem kann eine Nachricht auch vom automatisierten System mit einer Nachricht beantwortet werden. Hierzu können in der Ontologie die Filterparameter für den dem Interpreter vorgeschalteten Filter definiert werden. Dadurch ist es möglich Informationen auszublenden, welche für andere Teilnehmer des Busses bestimmt sind. Diese Filterparameter können allgemeiner Natur sein, wenn es möglich ist die Nachrichten für das Gateway an generellen Merkmalen zu identifizieren (z.B. bei einem CAN-Bus an einem speziellen Identifier). Oder spezifischer Natur, wenn zu einer Operation eine Antwort des automatisierten Systems gehört und diese anhand spezifischer Merkmale (z.B. bekannte Teile der Nachricht) identifiziert werden kann.

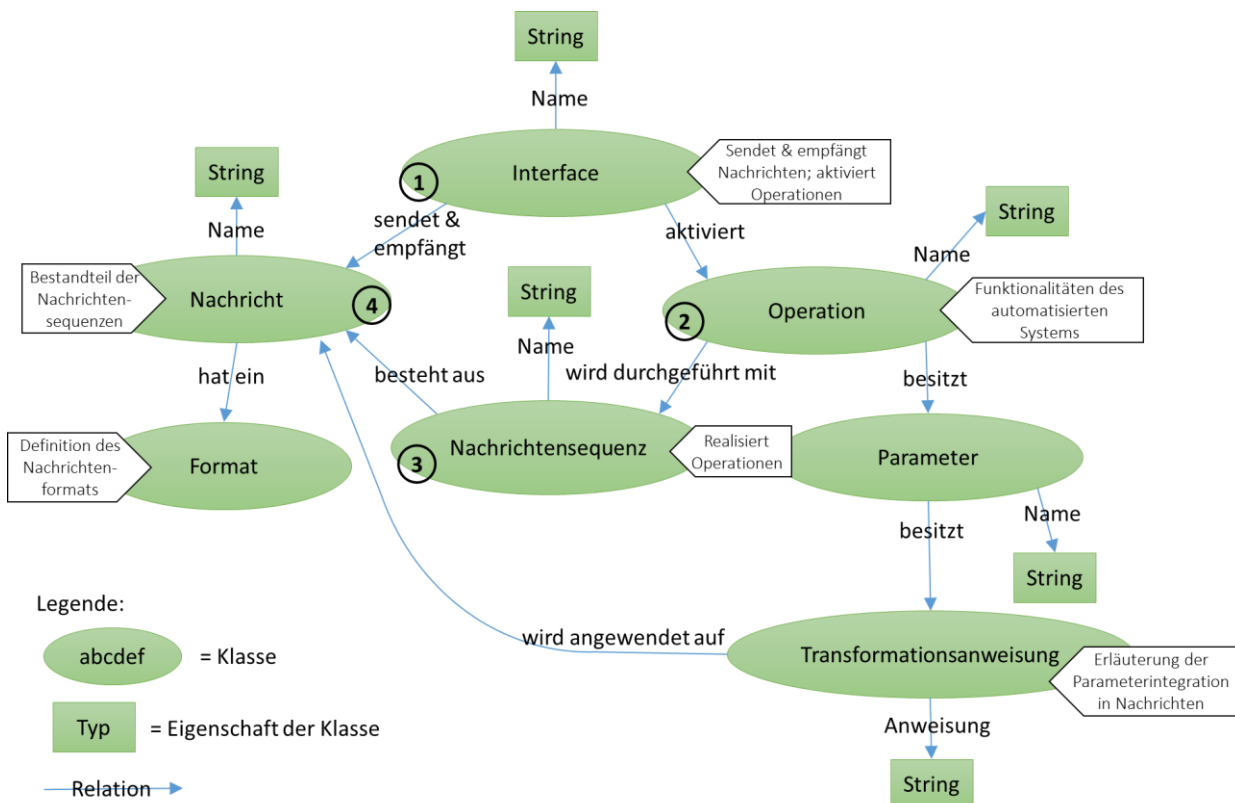


Abbildung 2: Ontologie der allgemeinen Beschreibung

Ein einfaches Beispiel für die oben genannten Bestandteile ist das Interface einer Kaffeemaschine (Punkt 1), welche die Operation „Kaffee produzieren“ durchführen kann (Punkt 2). Hierfür muss

eine Nachrichtensequenz (Punkt 3) bestehend aus einer einzelnen Nachricht „Kaffee“ (Punkt 4) an die Kaffeemaschine gesendet werden.

### **3.2 Der Interpret**

Der Interpret stellt die Verbindung zwischen Gateway und Bussystem dar. Die Hauptaufgabe des Interpreters ist es, die Nachrichten des Gateways bzw. des automatisierten Systems zu analysieren. Hierzu muss er die Bestandteile extrahieren und zusammen mit den Informationen der Ontologie in eine Form bringen, welche dann weitergeleitet wird. Da innerhalb eines automatisierten Systems über den Bus viele Nachrichten ausgetauscht werden, welche für das Gateway uninteressant sind, ist dem Interpret ein Filter vorgeschaltet, welcher diese Nachrichten nicht durch lässt. Die Information nach welchen Merkmalen gefiltert werden soll, ist ebenfalls in der Ontologie gespeichert.

## **4 Realisierungsbeispiel**

Das vorgestellte Konzept einer flexiblen Gatewayschnittstelle wurde prototypisch an einem Gastronomie-Kaffeefollautomaten umgesetzt, welcher über eine CAN-Schnittstelle verfügt. Zur Erstellung der Ontologie wurden folgenden Informationen über die Schnittstelle verwendet:

- das Kommunikationsprotokoll der Schnittstelle und der damit verbundenen Aufbau der Nachrichten
- die Operationen, welche vom Kaffeefollautomaten durchgeführt werden können
- die für die Durchführung der Operationen notwendigen Nachrichten und deren Reihenfolge
- möglichen Parameter der Operationen und wie diese in die Nachrichten integriert werden.

Zu den Operationen des Kaffeefollautomaten gehören unter anderem die Zubereitung eines Kaffees mit individueller Menge an Kaffeepulver und Wasser (`getCoffee`), die Bereitstellung von heißem Wasser für Tee (`getWater`) und das Auslesen der hinterlegten Rezepturen für unterschiedliche Kaffeevarianten (`getRecipe`).

Die vorgegebene Grundstruktur der Ontologie (siehe Abbildung 2) wurde zunächst um die Beschreibung des Nachrichtenformats, welche in Abbildung 3 dargestellt ist, erweitert. Hierbei handelt es sich um eine allgemeine Definition, welche für andere automatisierte Systeme mit einem CAN-Bus wiederverwendet werden kann. Anschließend wurden Instanzen der Klassen gebildet, welche die notwendigen Werte für die Eigenschaften besitzen. Abbildung 4 zeigt einen Ausschnitt der Instanzen des Kaffeefollautomaten, mit den von den Klassen definierten Beziehungen und den Eigenschaftswerten. Zusätzlich zur Ontologie wurde der Interpret und eine Benutzungsschnittstelle realisiert, welche es ermöglicht mittels einfachen XML-basierten Nachrichten verschiedene Operationen des Kaffeefollautomaten auszuführen.

Als Beispiel für die Durchführung einer Operation wird im Folgenden gezeigt, welche Schritte notwendig sind, um einen schwarzen Kaffee, bestehend aus 50ml Wasser und 15g Pulver, herzustellen. Hierzu sind unter anderem die Instanzen der Klassen der Ontologie in Abbildung 4 notwendig, welche im Folgenden genauer erläutert werden.

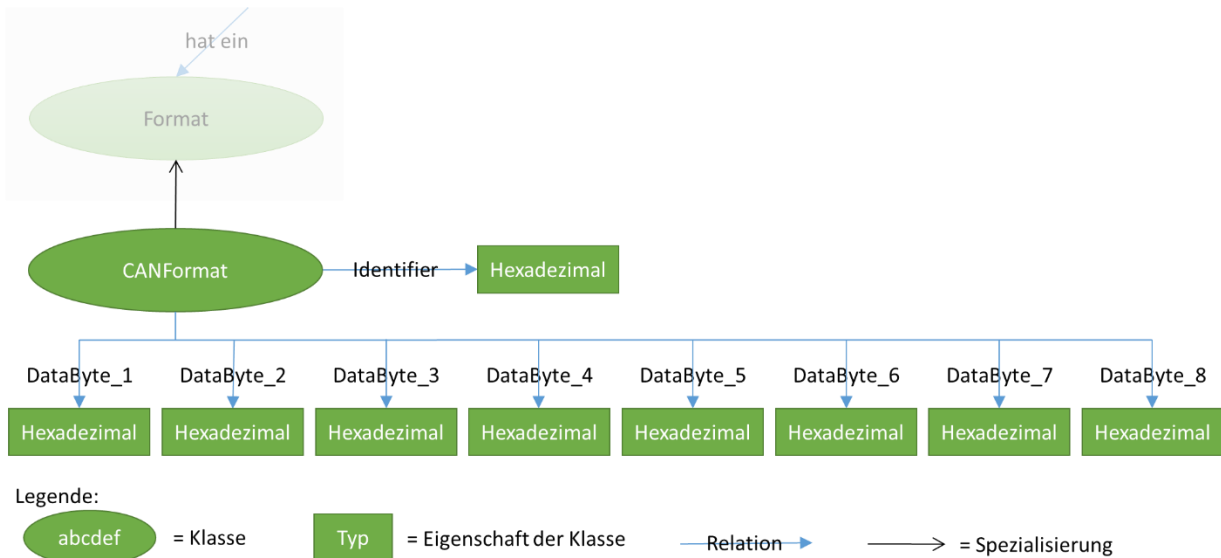


Abbildung 3: Verfeinerung der allgemeinen Beschreibung für den Kaffeevollautomaten

Eingeleitet wird der Prozess durch eine entsprechende XML-Nachricht an den Interpretier. Diese wird vom Interpretier in ihre Teile aufgeteilt. In diesem Beispiel sind dies die gewünschte Operation („GetCoffee“) und die Liste der Parameter („Water“ und „Powder“) mit ihren jeweiligen Werten (50 bzw. 15). Im nächsten Schritt werden vom Interpretier die Informationen zur Operation „GetCoffee“ aus der Ontologie abgefragt. Hierzu startet die Analyse beim Interfaceobjekt I14, von wo aus das Operationsobjekt I1 mit dem Name „getCoffee“ erreicht wird.

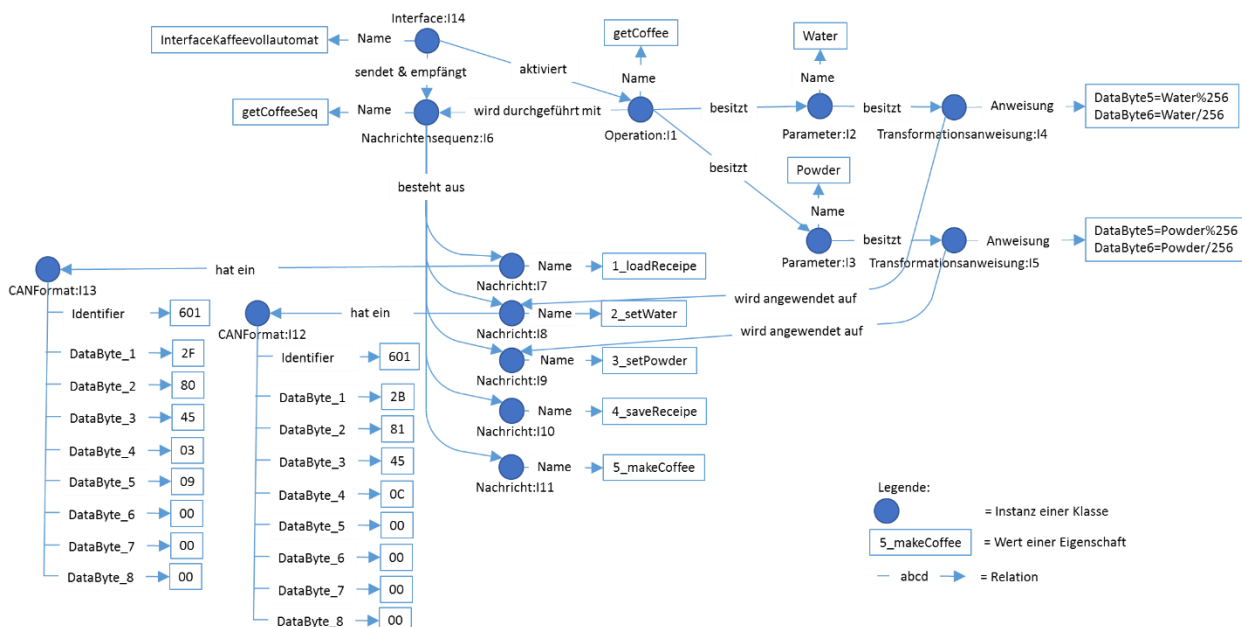


Abbildung 4: Auswahl der Instanzen der Ontologie des Kaffeevollautomaten

Dieses hat eine Beziehung zur Nachrichtensequenz I6, welche eine Beziehung zu den Nachrichtenobjekten I7 – I11 hat. Jedes dieser Nachrichtenobjekte hat eine Beziehung mit einem Objekt der Klasse CANFormat, welche das Format einer CAN-Nachricht definiert. Zusätzlich dazu hat I1 Beziehungen zu den Parameter-Objekten I2 und I3 mit den Namen „Water“ und „Powder“, welche wiederum eine Beziehung zu einer Transformationsanweisung besitzen. Diese Anweisung beschreibt, auf welche Weise der Parameterwert in eine Nachricht integriert werden muss. Dies ist zum Beispiel bei der Nachricht „2\_setWater“ so, dass für den Wert des 5. Datenbyte die Berechnungsanweisung „Water%256“ angegeben ist, bzw. für das 6. Datenbyte die Anweisung „Water/256“. Diese Anweisungen bedeuten, dass der Wert des 5. Datenbytes sich aus dem Wert des Parameters Water modulo 256 ergibt. Der Wert des 6. Datenbytes ergibt sich aus der Ganzzahldivision des Parameterwertes durch 256, womit der Wert des Parameters in zwei 8 Bit lange Zahlen aufgeteilt wird. Für die Nachricht „2\_setWater“ ergeben sich somit die Werte: 0x601, 0x2B, 0x81, 0x45, 0x0C, 0x50, 0x00, 0x00, 0x00. Abgeschlossen wird die Aktion dadurch, dass die Nachrichten, entsprechend ergänzt mit den jeweiligen Parameterwerten, über den CAN-Bus an den Kaffeefullautomaten gesendet werden, wodurch dieser zur Produktion des gewünschten Kaffees veranlasst wird.

## **5 Fazit**

Das in diesem Beitrag vorgestellte Konzept einer flexiblen Gatewayschnittstelle stellt eine Möglichkeit dar, ein Gateway an unterschiedliche Bussysteme anzupassen, ohne die eigentlichen Systemsoftware des Gateways anzupassen. Dies wird dadurch erreicht, dass in einer Ontologie die Eigenschaften der Schnittstelle beschrieben werden und es dadurch möglich ist unterschiedliche automatisierte Systeme auf eine einheitliche Weise anzusprechen. Hierbei können mit der Ontologie auch komplizierte Kommunikationsabläufe beschrieben werden, welche mehrere Nachrichten und Parameter benötigen.

Validiert wurde das Konzept anhand eines Gastronomie-Kaffeefullautomaten mit einer CAN-Bus Schnittstelle. Hierbei wurde gezeigt, dass es mit dem Konzept möglich ist mit dem automatisierten System zu kommunizieren und verschiedene Kommunikationssequenzen auszuführen. Außerdem konnte gezeigt werden, dass neue Funktionalitäten durch ein einfaches Update der Ontologie zur Laufzeit bereitgestellt werden können.

## **6 Ausblick**

Zur Untersuchung des Aufwandes bei der Realisierung eines zusätzlichen Bussystems soll das Konzept auf einen Waschtrockner mit integriertem Bussystem angewendet werden. Hierdurch wird auch die Anpassbarkeit des Konzeptes evaluiert.

Zudem wird das Konzept der flexiblen Gatewayschnittstelle in ein Konzept zur Erweiterung bestehender automatisierter Systeme [FaGö15] integriert. Dieses Konzept ermöglicht es bestehende automatisierte Systeme in Industrie-4.0-Szenarien zu integrieren, indem die automatisierten Systeme von einem Konnektor in den Szenarien vertreten werden. Dieser

Konnektor benutzt die vorgestellte flexible Gatewayschnittstelle zur Kommunikation mit dem automatisierten System über eine der vorhandenen Schnittstellen von diesem.

## 7 Literatur

- [Bräut2011] Bräutigam, C.: Schnittstelle flexible realisieren mit FPGA, etz, Heft 8/2011, Seiten 2-4, 2011, [http://www.etz.de/files/e10822zfe\\_softing.pdf](http://www.etz.de/files/e10822zfe_softing.pdf), Zugegriffen: 20. November 2015
- [ETSI2013] ETSI: Machine-to-Machine communications (M2M); Functional architecture, ETSI TS 102 690 V2.1.1, 2013  
[http://www.etsi.org/deliver/etsi\\_ts/102600\\_102699/102690/02.01.01\\_60/ts\\_102690v020101p.pdf](http://www.etsi.org/deliver/etsi_ts/102600_102699/102690/02.01.01_60/ts_102690v020101p.pdf), Zugegriffen: 25. November 2015
- [FaGö15] Faul, A.; Göhner, P.: Framework für die dynamische Kooperation von automatisierten Systemen. In: Automation 2015. 16. Branchentreff der Mess- und Automatisierungstechnik, 11. und 12. Juni 2015, Baden-Baden, (VDI-Berichte), Düsseldorf: VDI-Verlag, 2015
- [IIC2015] IIC: Industrial Internet Reference Architecture, Version 1.7, 2015  
<https://www.iiconsortium.org/IIRA.htm>, Zugegriffen: 20. November 2015
- [Plat2015] Plattform Industrie 4.0: Normen & Standards, 2015, <http://www.plattform-i40.de/I40/Navigation/DE/Industrie40/Handlungsfelder/NormenStandards/normen-und-standards.html>, Zugegriffen 6. Dezember 2015
- [RFC0196] RFC 196: Device Abstraction Layer, Final, 2014,  
<https://github.com/osgi/design/raw/master/rfcs/rfc0196/rfc-0196-DeviceAbstractionLayer.pdf>, Zugegriffen: 25. November 2015
- [RMZ2014] Riedl, M.; Zipper, H.; Meier, M.; Diedrich, C.: Cyber-physical systems alter automation architectures, In: Annual reviews in control. - Amsterdam [u.a.] : Elsevier Science, Bd. 38.2014, 1, S. 123-133
- [VDB2013] Vogel-Heuser, B.; Diedrich, C.; Broy, M.: Anforderungen an CPS aus Sicht der Automatisierungstechnik. In: Automatisierungstechnik (at), Vol. 61, No. 10, 2013, S. 669–676
- [VDI2015] VDI/VDE: Statusreport Referenzarchitekturmodell Industrie 4.0 (RAMI4.0), 2015, [https://www.vdi.de/fileadmin/user\\_upload/VDI-GMA\\_Statusreport\\_Referenzarchitekturmodell-Industrie40.pdf](https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf), Zugegriffen: 20. November 2015