

Software-Komponentenmodell für ein multikonnectives Maschinen-Kommunikationsmodul

Arndt Kritzner¹ und Jens Teichmann²

Abstract: Im Folgenden wird das Software-Systemkonzept eines Maschinen-Kommunikationsmoduls vorgestellt, das sowohl für Dokumentationszwecke als auch zur Fahrereinformation, für kooperative Mehrmaschinenarbeit und für maschinentechnische Belange Anwendung findet. Die verfügbaren Leistungsmerkmale und die in der Anwendung gesammelten Erfahrungen werden diskutiert.

Keywords: Kommunikationsmodul, Softwarestack, Mobile Edge Cloud

1 Ausgangssituation

Vor dem Hintergrund immer stärkerer Vernetzung und digitaler Verflechtung des Landmaschinenbetriebes müssen die Softwarebestandteile des Maschinen-Kommunikationsmoduls immer umfangreichere Aufgaben abbilden und Datenströme bewältigen. Kommunikationstechnisch finden aus stationären Installationen bekannte Technologien wie Ethernet und Webschnittstellen immer mehr Anwendung auf mobilen Maschinen [BM16]. Um den wachsenden Anforderungen herstellerübergreifend, flexibel und sicher gerecht zu werden, ist die Anwendung einer strukturierten Softwareumgebung mit für den landwirtschaftlichen Anwendungsfall optimierten Eigenschaften von Vorteil.

2 Anwendungskontext

Vom Maschinen-Kommunikationsmodul wird eine lokale Netzwerkzelle aufgebaut, in die über verschiedene verdrahtete und drahtlose Schnittstellen unterschiedliche Peripheriekomponenten und Teilnehmer der lokalen Umgebung auf dem Feld einbezogen werden. Im realen Praxisbetrieb kann von einer lückenlosen Mobilfunkabdeckung für die Kommunikation zu zentralen Cloud-Instanzen nicht ausgegangen werden. Deshalb muss ein wesentlicher Teil der Systemintelligenz lokal direkt vom Kommunikationsmodul bereitgestellt werden, um eine kontinuierliche Systemfunktion zu gewährleisten [HLJ16].

Aus den Verwendungszwecken der Daten ergeben sich zwei abzubildende Bereitstellungscharakteristiken:

¹ Logic Way GmbH, Hagenower Straße 73, 19061 Schwerin, kritzner@logicway.de

² TU Dresden, Professur für Agrarsystemtechnik, 01069 Dresden, teichmann@ast.mw.tu-dresden.de

- verzögerungsarme Bereitstellung von Momentanwerten als kontinuierlicher Datenstrom für mitlaufende Prozessinformationen. Aktualität hat Vorrang vor der Lückenlosigkeit der Übertragung.
- asynchrone Datenbereitstellung für nachlaufende Dokumentations-, Analyse- und Abrechnungszwecke. Die Lückenlosigkeit der übertragenen Daten ist maßgeblich und muss auf Empfängerseite überprüfbar sein.

Trotz eventuell momentan ausreichender Übertragungsbandbreite kommt der adaptiven Filterung von Dateninhalten eine entscheidende Bedeutung zu, um Übertragungsbandbreite, Datenverarbeitungsleistung und Kommunikationskosten sinnvoll zu begrenzen. Die Klassifikation der Relevanz bestimmter Datentelegramme ist dafür die Grundlage [Ed18].

3 Software-Komponentenmodell

Das Software-Systemkonzept eines herstellerübergreifenden Maschinen-Kommunikationsmoduls wurde so ausgelegt, dass sowohl Dokumentationszwecke als auch Momentanbelange wie Fahrerinformation und -interaktion, kooperative Mehrmaschinenarbeit und maschinentechnische Unterstützung abgebildet werden können.

Die Softwarekomponenten decken anwendungsneutral alle technischen Erfordernisse der Datenerfassung und -kommunikation ab und stellen ebenfalls eine repräsentative Auswahl an logischen Bausteinen für typische Aufgabenstellungen im landwirtschaftlichen Technikeinsatz zur Verfügung.

Die Aufteilung der Softwarekomponenten (Tab. 1) wurde in vier Schichten vorgenommen, die über ZeroMQ-Datensammelschienen untereinander verknüpft sind.

Die Basisfunktion „nachlaufender Datentransfer“ wird bereits in der Sachdienste-Schicht abgebildet und ist nicht auf zusätzliche Apps angewiesen.

Die Softwareinstallation des Kommunikationsmoduls basiert auf einem speziell konfigurierten Linux-Grundsystem mit einer Zusammenstellung generischer Komponenten, auf die eine Schicht speziell für den landwirtschaftlichen Kontext entwickelter anwendungsübergreifender Bausteine aufsetzt. Die darüber liegende Schicht der anwendungsfallbezogenen „Apps“ wird über eine Struktur aus ZeroMQ-Datensammelschienen und Sockets mit Maschinendaten versorgt. Für die Globalkommunikation zu Cloud-Instanzen wird eine Kommunikationskomponente („Gatekeeper“) eingesetzt, die die Gesamtkommunikation über als zulässig vorgegebene Verbindungen abwickelt. Für die Fahrerinformation und -interaktion über ein Smart Terminal wird ein lokaler Webserver mit serverseitigem Lua- und clientseitigem Javascript und dem Web-Framework JQuery bereitgestellt. Webfähige Peripheriekomponenten greifen über REST-Schnittstellen auf die gleiche Serverinstanz zu. Über den vom Kommunikationsmodul

bereitgestellten DNS-Nameservice wird für mehrere virtuelle Hosts ein einheitlicher URL-Namensraum aufgespannt.

Logische Schicht	Funktionalität	Komponenten (Auswahl)
Systemdienste	Bereitstellung und Vereinheitlichung des Zugangs zu physikalischen Schnittstellen (CAN, GNSS, LAN, WLAN, WAN etc.)	cansocket, gpsd, Nginx, sshd, DNS, DHCP, NATS Server, Azure SDK, Sqlite, libCurl etc.
Sachdienste	Bereitstellung wiederverwendbarer Basisfunktionalität mit logisch-dateninhaltlichem Bezug zum eigentlichen Produktivprozess (Datenkonzentration, Filterung, Warteschlangenmanagement, digitales Umgebungsmodell, Authentifikation, asynchroner Datentransfer)	DCS (Datenkonsolidierungsdienst), PlugSense/ISOplug (Anbaugeräteerkennung), SiteSense (Umfeldabbildung), AsyncTransfer, SmartTerminal, SiFiLib (Situations-Fingerprinting) etc.
Sachanwendungen (Apps)	Anwendungsfallbezogene Programmbestandteile (z. B. Logistik, Abrechnung, Nachverfolgung, Firmware-Management, Service-Zugang, Parametrierung)	
Datenlink	Management der Verbindungen zwischen Apps und Cloud-Instanzen unter Berücksichtigung der vorgegebenen Kommunikationsreglementierung	GateKeeper

Tab. 1: Softwareschichten und deren Ökosystem

Die Kommunikationskomponente „GateKeeper“ kommuniziert über HTTP/REST bzw. MQTT (zusätzlich AMQP zu Microsoft Azure) mit Zugangspunkten in der Cloud. Die GateKeeper-Kommunikation kann per Konfiguration auf eine Liste von erlaubten Cloud-seitigen Kommunikationsendpunkten eingeschränkt werden. In dieser weißen Liste nicht vorgesehene Kommunikation wird über den hinterlegten Standard-Kommunikationsendpunkt umgeleitet und kann dort weiterverteilt und ggf. abgerechnet werden

4 Anwendungsübergreifende Funktionalität

Die Vielfalt der umsetzbaren Anwendungsfälle im landwirtschaftlichen Produktionsprozess nutzt in hohem Maße dieselben Mechanismen und deshalb

zweckmäßigerweise auch dieselben Softwareressourcen. Die im Software-Komponentenmodell angeordneten wiederverwendbaren Funktionsblöcke für Datentransport, -aggregation, -filterung, Umgebungsmodellbildung und ein Interaktionsterminal können deshalb einen wesentlichen Teil fast aller Anwendungsfall-Lösungen bilden, der dann jeweils nur noch um eine spezielle, relativ leichtgewichtige Anwendungslogik ergänzt werden muss. Nutzerprofilabhängige Datenströme können so effizient erzeugt werden.

Um redundante Datenkommunikation zu vermeiden, sind in grundlegenden Datenaggregationsdiensten in der vom Modul bereitgestellten Mobile Edge Cloud bereits relevanzadaptive Filtermechanismen für Geo- und Sachdaten verfügbar, die die wiederholte Übertragung gleicher oder zu ähnlicher Dateninhalte wirksam reduzieren. Über unterschiedliche Kommunikationsmechanismen werden Prozesse mit unterschiedlichen Echtzeitanforderungen bedient.

5 Codeeffizienz

Die Umsetzung einer strukturierten Softwareumgebung mit definierten Schnittstellen und Interprozess-Kommunikationsmechanismen führt, solange nur ein einziger Anwendungszweck verfolgt wird, zu erhöhtem Programmieraufwand und Codeumfang. Durch die Wiederverwendbarkeit der Funktionsbausteine für unterschiedliche Anwendungsaufgaben kehrt sich dieser Nachteil allerdings schnell um, das einzelne Codesegment wird besser ausgelastet und vorhandener Funktionsumfang wird parallel für unterschiedliche Anwendungszwecke genutzt [Gö98].

Für den Datenfluss von der physikalischen Schnittstelle – beispielsweise CAN-Bus – bis zur Übertragung der Daten zum Cloud-Service werden in der gewählten Architektur 3 bis 5 Interprozess-Schnittstellen passiert. Auf die Einzelanwendung betrachtet entfallen ca. 10-20 % des Funktionsumfangs auf Schnittstellenfunktionalität. Bereits durch die Verwendung desselben Funktionsbausteins für zwei Anwendungsaufgaben wird eine kompaktere Umsetzung gegenüber durchgehender Implementierung innerhalb einer Anwendung erreicht.

6 Sicherheit, Testbarkeit

Die Zerlegung der Gesamtfunktionalität in logische Blöcke, die über definierte Schnittstellen kommunizieren (ZeroMQ, Sockets, MQTT, REST), ermöglicht die feingranulare Reglementierung von Anwendungsberechtigungen bereits mit Betriebssystemmitteln (SE-Linux). Die klar abgegrenzte Funktionalität pro Block bietet andererseits die Möglichkeit, den Funktionsumfang mit hoher Abdeckung separat zu testen und dadurch das regelkonforme Funktionieren zu gewährleisten. Die im planmäßigen Betrieb genutzten Schnittstellen bilden dabei auch die Einleitungs- und Abgreifpunkte für Teststimulation und Ergebnisüberprüfung.

Die Gerätesicherheit der Datenverarbeitung auf dem Kommunikationsmodul stützt sich auf eine X509-Public-Key-Infrastruktur und die grundsätzliche Verwendung verschlüsselter Verbindungen in der IP-Kommunikation über Gerätegrenzen hinweg. Alle sicherheitsrelevanten Merkmale werden dabei im Inbetriebnahmeablauf vom Modul selbst erzeugt und sind unikat.

Für die Durchsetzung von Datensicherheit und gleichzeitige sachbezogene automatisierte Authentifikation von Datenabgabe-Anfragen wird Situations-Fingerprinting als Bibliothek zur Verfügung gestellt [KT18]. Die Modellierung des Zusammenwirkens unterschiedlicher Akteure und Geräte auf dem Feld kann dadurch in hohem Maße automatisiert stattfinden, wobei durch feingranulare Datenportionierung auch nur der tatsächlich sachbezogen veranlasste Datenausschnitt ausgetauscht wird.

7 Anwendbarkeit

Bisher umgesetzte Anwendungsfälle umfassen beispielsweise Dokumentationszwecke, Maschinenbetriebsüberwachung und Service-Unterstützung ebenso wie Firmware-Update-Management für das Maschinengespann.

Naheliegender ist die Umsetzung von Anwendungsfällen aus den Bereichen Feld- und Transportlogistik, Maschinenbetrieboptimierung, Entscheidungsunterstützung, Produktnachverfolgung, Aufgabenplanung und weiteren, die ebenfalls zweckmäßigerweise größere Funktionsblöcke in das dargestellte Komponentenmodell auslagern könnten.



Abb. 1: Maschinen-Kommunikationsmodul, auf dem der Softwarestack implementiert und erprobt wurde (CPU: ARM Cortex A8, 2xCAN, LAN, LTE, Buildroot Linux)

8 Konkrete Referenzumsetzung

Das beschriebene Software-Komponentensystem für ein in Landmaschinen genutztes Kommunikationsmodul wurde in Betrieb genommen und bereits im Feld für eine Ernteperiode mit mehreren Maschinen erprobt. In dieser Phase wurden Daten von Mähreschern, Feldhäckslern, Traktoren, LKWs und Güllefahrzeugen erfasst und aufbereitet.

Zur Einbindung in die Bordelektronik und Kommunikationsstrukturen von Landmaschinen verfügt das in Abb. 1 gezeigte Kommunikationsmodul über eine intelligente Platinensteuerung, die Zündungssignal und alle internen Stromschienen überwacht sowie das Latentenergie-Management übernimmt. Der Inhalt des Latentenergiespeichers wird genutzt, um bei abruptem Stromversorgungsverlust noch einen geordneten Systemabschluss durchzuführen.

9 Ausblick

Bisher wurde die beschriebene Software-Komponentenstruktur im produktiven Dauerbetrieb, allerdings nur im Rahmen wissenschaftlich-technischer Forschungsvorhaben eingesetzt [Do18] [Bi18]. Um Anwendungshemmnisse zu beseitigen und eine breitere Nutzung zu stimulieren, werden die Komponenten als Open Source Software entwickelt.

Für nichtkommerzielle Verwendung ist die Nutzung kostenlos.

Literaturverzeichnis

- [Bi18] Forschungsprojekt „BiDa-LaP“, www.bidalap.de, Stand: 06.12.2018.
- [BM16] BMEL: Tier und Technik, Landwirtschaft verstehen, Im Fokus: Chancen der Digitalisierung, Bundesministerium für Ernährung und Landwirtschaft, S. 8 ff., 2016.
- [Do18] Forschungsprojekt „DoHLe“, www.dohle-tnt.de, Stand: 06.12.2018.
- [Ed18] Edge Computing Task Group: Introduction to Edge Computing in IIoT, Edge Computing Task Group, S. 6 ff., 2018.
- [Gö98] Göhner, P.: Komponentenbasierte Entwicklung von Automatisierungssystemen. In: GMA-Kongress Mess- und Automatisierungstechnik. VDI-VDE-Gesellschaft Mess- und Automatisierungstechnik, Ludwigsburg, 1998.
- [HLJ16] Heidrich, M.; Luo; Jijun, J.: Industrial Internet of Things: Referenzarchitektur für die Kommunikation, Fraunhofer ESK, S. 19 ff., 2016.
- [KT18] Kritzner, A.; Teichmann, J.: „Situations-Fingerabdruck“ - Verwaltungsstrukturübergreifendes automatisiertes Berechtigungsmanagement für landwirtschaftliche Daten nach sachbezogenen Kriterien. In: 38. GIL-Jahrestagung. Gesellschaft für Informatik e.V., Kiel, S. 143-146, 2018.