Multimodal Task Assignment and Introspection in Distributed Agricultural Harvesting Processes

Daniel Porta*, Zeynep Tuncer†, Michael Wirth‡ and Michael Hellenschmidt§
*German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany, daniel.porta@dfki.de
†John Deere GmbH & Co. KG, Kaiserslautern, Germany, TuncerZeynep@JohnDeere.com
‡Eyeled GmbH, Saarbrücken, Germany, wirth@eyeled.de
§SAP AG, Darmstadt, Germany, michael.hellenschmidt@sap.com

***Abstract*—Harvesting processes are in fact industrial manufacturing processes that follow a tight schedule. Unexpected incidents can disturb a harvest and require a replanning of the process in order to avoid severe financial losses. When a new plan has been found, it must be communicated to the affected process participants, i.e., drivers of agricultural machinery. This paper presents a cloud-based system for orchestrating and coordinating a fleet of agricultural machinery and their drivers during an ongoing harvest in case of an unexpected incident. A management dashboard allows the real-time replanning of a harvesting process and sends updated instructions to each affected driver's mobile device. The paper focuses on the communication between the contractor and a driver in the field as well as the interaction of the driver with his mobile device. It is explained how the system accomplishes a fast, traceable, and safe communication with the drivers that may suffer from bad network conditions and a high cognitive load. In order to understand the details of his new tasks, a driver can examine them in a multimodal dialogue including speech with the system. This is beneficial in a driving situation. By interacting with the mobile client, the system also deduces if the driver correctly understood his new instructions and can intervene if not.**

***Keywords*–Multimodal Dialogue; Task Assignment; Task Introspection; Agriculture; Harvesting Process.**

I. Introduction

Because of its complexity, agricultural production can nowadays be regarded as an industrial manufacturing process, in which a tractor in a sense represents not merely a vehicle but a complex tool that is, due to the size of the production area, mobile by means of wheels. The interests of agricultural enterprises are also comparable to other production companies. On the one hand, the entrepreneur, i.e., the farmer, wants to design process chains that achieve the highest possible efficiency. On the other hand, these production processes should to some extend be robust against possible interference. Thus, the designed processes have significant effects on the efficiency of the machines, but also on the products themselves. Additionally, a particular focus in agriculture lies upon ecology and sustainability, which are also affected by the operating procedures. Considering a harvest campaign as a complex agricultural process chain, a large number of agricultural machinery, e.g., tractors and forage harvesters, is involved in the distributed process. Their coordination and cooperation must be perfectly organized to accomplish the harvesting process in an economical manner. They are based on a complex orchestration of all participating employees and machines which is planned before the actual harvest begins. The necessary interaction between the machines require process technologies as in any conventional factories. However, external and unexpected influences, such as changing weather or traffic conditions, may affect this sensitive structure as well as a drop out of a machine or driver.

A harvest is typically planned and conducted by a contractor on behalf of the farmers. A contractor is a service provider that has a large fleet of agricultural machinery and personnel at his disposal. In case of an unexpected incident that affects an ongoing harvest, the contractor gets in trouble. He normally serves multiple customers at different locations at the same time and his schedule for the remaining harvest season often leaves no room for catching up a bigger delay. The contractor, therefore, requires tools based on Information Technology (IT) for decision making and transmission of information in order to immediately change action plans for agricultural machinery and their drivers. Subsequently, all stakeholders need to be informed about the change of plan. This is done today often via mobile phones or radios. Telephony as a synchronous form of communication requires a time-consuming, sequential approach in order to inform the large number of affected drivers (we consider a smaller number of 20 drivers in total) and also suffers in rural areas from poor network coverage. Radios, however, often do not have the required range and are just not safe to use for inexperienced seasonal workers. Thus, some driver might follow the new plan while others still follow the outdated plan.

This paper presents a cloud-based harvest and communication management system for the COordination of Agricultural Production in Real-Time (COAP-RT). It is capable of orchestrating and coordinating a fleet of agricultural machinery and their drivers during an ongoing harvest. The paper focuses especially on the drivers' multimodal mobile User Interface (UI) for communicating changed action plans as a result of an unexpected incident. This communication must be *quick* and *traceable* for the contractor and *safe* for the driver. These three factors are important for keeping track of a tight schedule. Also, these factors become even crucial considering that tractor drivers suffer from a high cognitive load in phases of high concentration, e.g., (un-) loading procedures or driving over rather small roads with tons of cargo and opposing traffic. This leaves only little room for additional attention. In such situations, multimodal dialogue interaction including speech and gestures proved to be beneficial [1].

The outline of the paper is as follows. In the next Section II, related work is discussed. Section III covers the overall system architecture and describes the implemented components. In Section IV, a complete user interaction with the mobile client is discussed. Finally, we conclude in Section V.

 ISBN: 978-1-61208-353-7

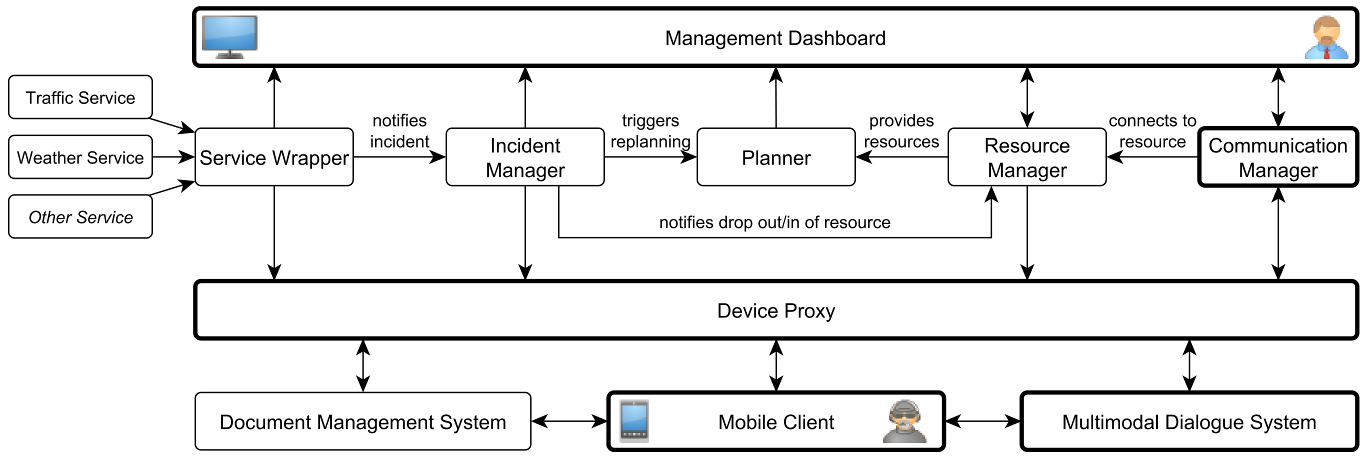

Figure 1. Overall logical system architecture. The highlighted components are relevant for the focus of this paper.

II. Related Work

There is an increasing number of IT-based research regarding precision farming or precision agriculture aiming at knowledge representation, information management, and decision support for various agricultural tasks that apply Future Internet technologies and concepts like the Internet of Services, the Internet of Things, the Semantic Web, and Industrial Internet [2][3].

The iGreen Project [4] developed an infrastructure for knowledge and service networks based on a service-oriented architecture in order to implement mobile decision support systems and tools for collecting and exchanging knowledge over organizational boundaries in the domain of agricultural production [5]. They use agroXML [6] and linked data technologies [7] in order to fit information systems to the requirements of agricultural processes. In essence, they propose the adoption of ReSTful [8] services because these are ”especially well suited to agriculture as they allow quick adaption to new conditions and reuse of data in unforeseen contexts” [9]. We also implemented a ReSTful service backend. The AGRICOLA project [10] developed an agent-based dynamic resource planning network in the agriculture domain. The planning considers weather and drop outs of personnel or machinery as dynamic disturbance factors. The planning itself relies on simulation-based dynamic coalition forming [11] in order to a achieve stable groups of cooperative agents. Our planning component is inspired by this approach. The Marion project [12] focuses on a dynamic and distributed infield planning system for harvesting. This planning considers route planning of (autonomous) vehicles within a field for optimal and smooth harvesting processes. This is a fine-grained micromanagement approach. In contrast to this, the COAP-RT system rather looks at the harvesting process from a bird's eye view. However, both approaches would perfectly fit together and complement each other. There are also commercial software-as-a-service platforms like Farmpilot [13] or 365FarmNet [14] available. Farmpilot can be seen as a fleet management system and harvest planning tool. However, it does not support automatic replanning of a process during an ongoing harvest campaign in case of an unexpected incident and the provided mobile application is only intended for use by the contractor, not the drivers. 365FarmNet aims at an open and holistic process management and service platform for the agricultural domain ranging from sowing to harvest. In this sense, the COAP-RT system can be seen as a specialized service for replanning an ongoing harvest campaign with an advanced multimodal mobile UI for communicating changed action plans to affected drivers. The Farming 4.0 initiative [15] of Deutsche Telekom and the German agricultural machinery manufacturer CLAAS tackle similar issues like we do. However, a coordinating higher instance in case of an incident is not intended. Also, the mobile client does not provide clear instructions but only data whose interpretation is up to the driver. The initiative mainly aims at testing communication technologies, such as Long-Term Evolution (LTE) and infield navigation in rural areas.

III. Overall System Architecture

This section describes the logical system architecture of the cloud-based COAP-RT system. Figure 1 depicts the respective components of the system and their connections. The highlighted components are relevant for the focus of this paper. However, the others are briefly presented in order to get a complete picture of the system. The cloud infrastructure relies on the SAP HANA Cloud platform [16].

The *Service Wrapper* makes data from external data sources internally available. Currently, we include the ReSTful traffic service from Microsoft [17] and the ReSTful weather service from OpenWeatherMap [18] which also provide weather forecast. Additional services can easily be integrated. The Service Wrapper polls these Web services in regular short intervals and republishes the data in an internal representation on a dedicated ReSTful application programming interface (API). Thus, the Service Wrapper acts as a Meta Web Service [19]. It also generates incidents based on rules if a process interfering situation is observed. Such an incident is propagated to the *Incident Manager*, which can also receive incidents from drivers created via their mobile client or even automatically generated incidents from their vehicles. For testing and demonstration purposes, incidents can be simulated. The Incident Manager then updates the remaining resources, e.g., in case a tractor dropped out, notifies the responsible contractor as well as affected drivers in the harvesting chain about the incident (that means they should prepare for forthcoming updates of their instructions) and triggers a replanning of the current harvesting process. The Incident Manager also keeps track of reported incidents. They can be retrieved from a dedicated ReSTful API. The replanning is carried out by the *Planner*. It takes available resources and other contextual constraints, such as deadlines and economical cost models, into account in order to come to an optimal solution. However, the replanning of a harvesting process is not an automatism. In the end, the contractor has to approve (or reject) the proposed replanned process. So, the contractor is always in charge and retains full control over the harvesting process, which will raise acceptance for IT-based tools in the experience-based agricultural domain. The Planner can also be accessed via a dedicated ReSTful API.

The *Management Dashboard* is the stationary UI for the contractor, similar to the Management and Monitoring Tool described in [20]. It is implemented as a browser-based mashup of available data and relevant information utilizing the ReSTful APIs of the system's back-end components. Thus, the contractor can initially enter, e.g., agricultural machinery at his disposal as well as customer records into the *Resource Manager*, a ReSTful component for managing master data. During a harvest campaign, he can get a quick overview of unexpected incidents that require his attention, affected customers, fields, and vehicles. In case a replanning becomes necessary, the contractor can revise countermeasures proposed by the Planner and, if he finally agrees, can broadcast a set of derived instructions to all affected drivers via the mobile Internet.

Communication with drivers is accomplished by the *Communication Manager*. When a driver signs in and connects to the system, this component links a virtual vehicle resource with the corresponding real-world vehicle. Afterward, the Communication Manager distributes messages addressed to the virtual resource to the actual driver's mobile client. It must be *traceable* for the contractor (1) when a message was send to a driver, (2) when it was received, (3) that a driver noticed the message and (4) that he understood the contents of the message. While the former two points tackle technical issues regarding network coverage in rural areas, the latter two points tackle important human factors because the contractor does not personally talk to the driver. When a driver has to concentrate on his actual work, he might not notice a new message or misunderstand (parts of) its content. If point (3) and/or (4) cannot be checked, it is very likely that drivers follow outdated, perhaps contradicting instructions and the replanned harvesting process is not executed as desired. Without asking the driver in such a situation, this cannot be revealed directly but may be derived later, e.g., by looking at position traces. In the next section, we will explain in depth how the COAP-RT system copes with the traceability of these four points. The Communication Manager does not communicate directly with the drivers' mobile clients but via the *Device Proxy*. Technically, both components can be merged. However, from a logical perspective, they serve different purposes. The Device Proxy is based on the lightweight Node.js framework and leverages the Session Initiation Protocol (SIP) [21] normally used by IP telephony. It implements a SIP registrar and proxy that manages a dictionary of SIP contact URIs and user names provided by the clients through SIP registrations. In this way, the Device Proxy can relay messages from the Communication Manager (received via HTTP) to the driver's actual mobile client (transmitted via SIP). This is by design of the SIP protocol efficient and to some extend robust against poor network coverage since the proxy uses the User Datagram Protocol (UDP) as SIP transport protocol. This communication is also easily traceable in a sense that the mobile client automatically acknowledges receipt of a message at the application layer.

The *Mobile Client* runs on a Google Nexus 7 with Android 4.2.2 or higher and is implemented as a cross-platform Apache Cordova application. It contains native Java plugins for SIP communication (we extended the source code of the stock Android SIP stack by instant messaging capabilities [22]), local speech synthesis, and bidirectional audio streaming in the common 8-bit A-law telephony codec. The graphical UI (GUI) as shown in Figure 2 is based on HTML5 and JavaScript, and thus, rendered in a Web browser component. It allows a driver to be always in touch with the most recent developments in the current harvest campaign. Pushing the red microphone button opens a channel for speech interaction. This function can also be triggered by pushing the call button of a connected bluetooth headset, since the client leverages vendor-specific headset events of the Android platform. The GUI consists of six interconnected views that are accessible after the driver signed in by providing his credentials and selecting a vehicle (Figure 2, screen (1)). This paper concentrates on the first two views. Most important, the *task view* (2) always explains the driver in a very clear and unmistakable fashion his current role and tasks within the harvesting chain. Tapping on a pin icon, the GUI switches to the map view where the corresponding target location is displayed. Incidents from the Indidents Manager are listed in the *incidents view* (3). The symbols indicate the source (from a vehicle, a weather, or traffic service), the severity, and the current status (unhandled, handled) of an incident. Tapping on a list entry reveals further details. A driver can report an incident by means of a dedicated form that is available via the menu button in the lower left corner of every screen. Neßelrath and Porta, [23] showed that reporting an incident can also be accomplished in a multimodal dialogue fashion. The *map view* (4) visualizes the driver's operational area and provides navigation aids to go there. The map view also visualizes the locations of relevant incidents that occurred during the harvest campaign. The *weather view* presents fine-grained current weather information and a three day forecast for the area of operation. This information comes from the Service Wrapper. The *vehicle view* displays selected vehicle specifications like dimensions and weights. This information comes from the Resource Manager and helps inexperienced drivers get acquainted more easily with the new vehicle. The *document view* allows the driver to access all manuals, regulations, and directions relevant for his current tasks as required by statutory law in PDF format. This is especially useful after a replanning occurred. In case a driver gets new instructions or is assigned to a different field which is now located in a landscape protection area or in a hillside situation, he does not need to return to the home base for picking up the required security advises in a paper-based form. These documents are provided by the context-sensitive *Document Management System*.

 ISBN: 978-1-61208-353-7

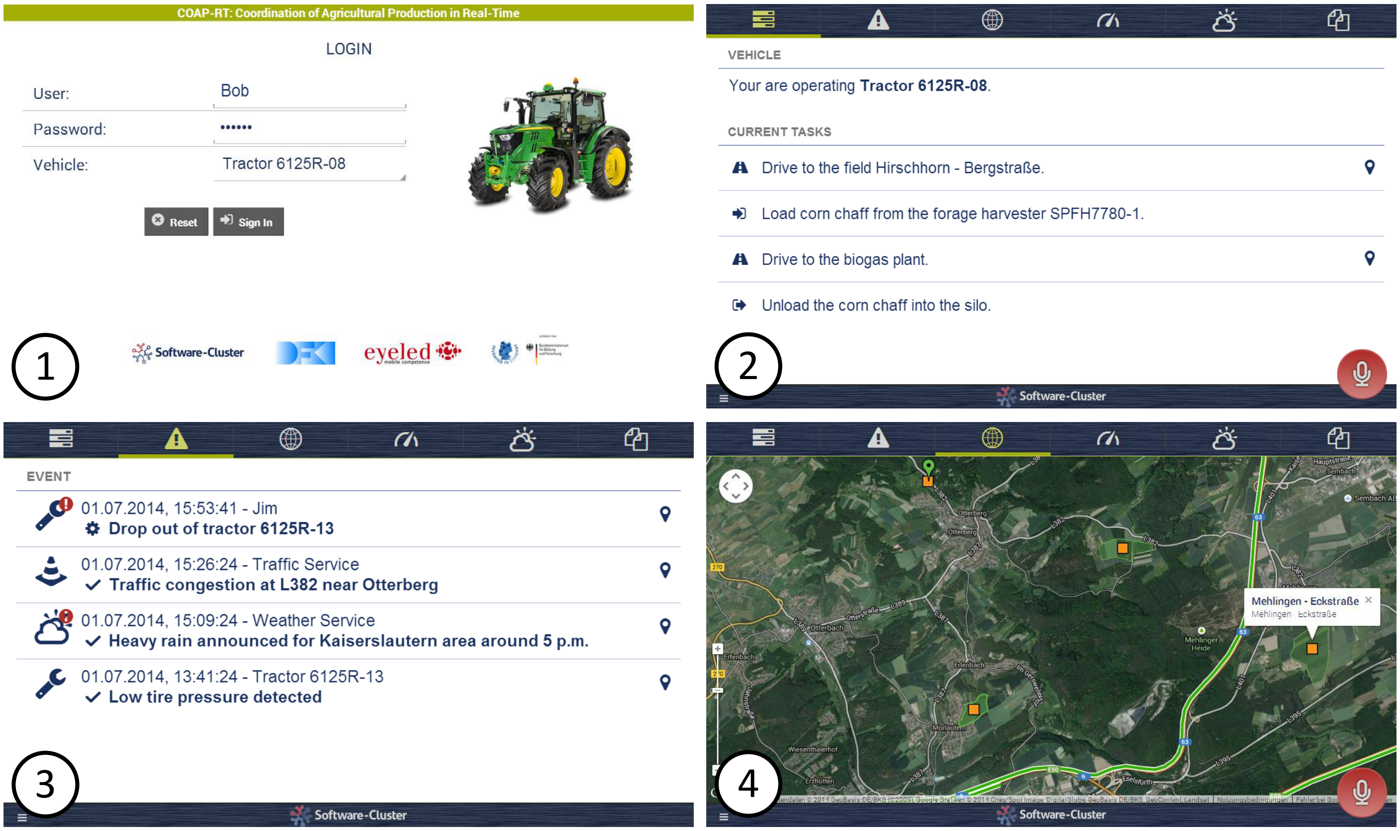

Figure 2. The GUI of the Mobile Client. Shown are the login screen (1), the task view (2), the incidents view (3), and the map view (4). The vehicle view, the weather view, and the document view are omitted for brevity. The paper focuses on the task view and the incidents view.

The *Multimodal Dialogue System* (MMDS) finally allows a driver to interact with the COAP-RT system in an advanced and non distracting fashion by using touch, speech, and other modalities. It is based on the SiAM-dp multitenant multimodal dialogue platform initially developed for the automotive domain [24]. It pursues a model-based development approach by means of the Eclipse Modeling Framework (EMF) [25] in order to build context-adaptive dialogue applications and it aims at considering the driver's cognitive load as one adaptation criterion. Consequently, SiAM-dp consists of a runtime environment and an Eclipse-based workbench for the rapid development of dialogue applications. Technically, the MMDS is also connected to the Device Proxy via SIP. When a driver signs in the COAP-RT system, the mobile client automatically creates a SIP session at the MMDS using the SIP INVITE procedure. Both components, the mobile client and the MMDS, can now communicate with each other by exchanging SIP messages. Eventually, this enables mixed-initiative conversations, i.e., either a driver or the COAP-RT system (by means of the MMDS) can start a conversation. The MMDS integrates off-the-shelf network speech recognition and synthesis solutions by means of SIP and MRCP [26]. Also, the MMDS implements a dialogue strategy, i.e., a strategy for leading a conversation with the driver, such that the contractor can be confident that the driver understood his new instructions. In this way, the MMDS can actually simulate a phone call with the driver on behalf of the contractor. Thus, the driver does not need to take his eyes from the street and can adapt to his current cognitive load.

IV. Multimodal Task Assignment and Introspection

As already mentioned in the introduction, the transmission of updated instructions from the contractor to the driver must be *quick*, *traceable*, and *safe*.

The transmission is *quick* as the MMDS can lead several conversations simultaneously. Thus, the COAP-RT system is faster than calling each affected driver individually and manually in sequence. If only one driver has to be instructed (which is unlikely in complex harvesting processes), the reaction time of the system still outperforms the contractor's reaction time, who might have to first search the correct phone number by hand. The system relies on the mobile Internet as the actual communication channel and improvements on the network infrastructure in rural areas are out of our scope. However, speech interaction requires less bandwidth than a normal phone call because the same audio codec is used for server-side speech recognition. Speech synthesis is performed directly on the mobile device. So, only the textual content needs to be transmitted from the MMDS to the client. If a driver's mobile device has no signal, he cannot be reached neither automatically by the system nor manually by the contractor. However, the system immediately recognizes when the driver is back online due to the device's renewed SIP registrations.

Regarding the technical *traceability* of the successful delivery of messages, we utilize the SIP protocol. So, the Communication Manager logs the timestamps of when a new message

 ISBN: 978-1-61208-353-7

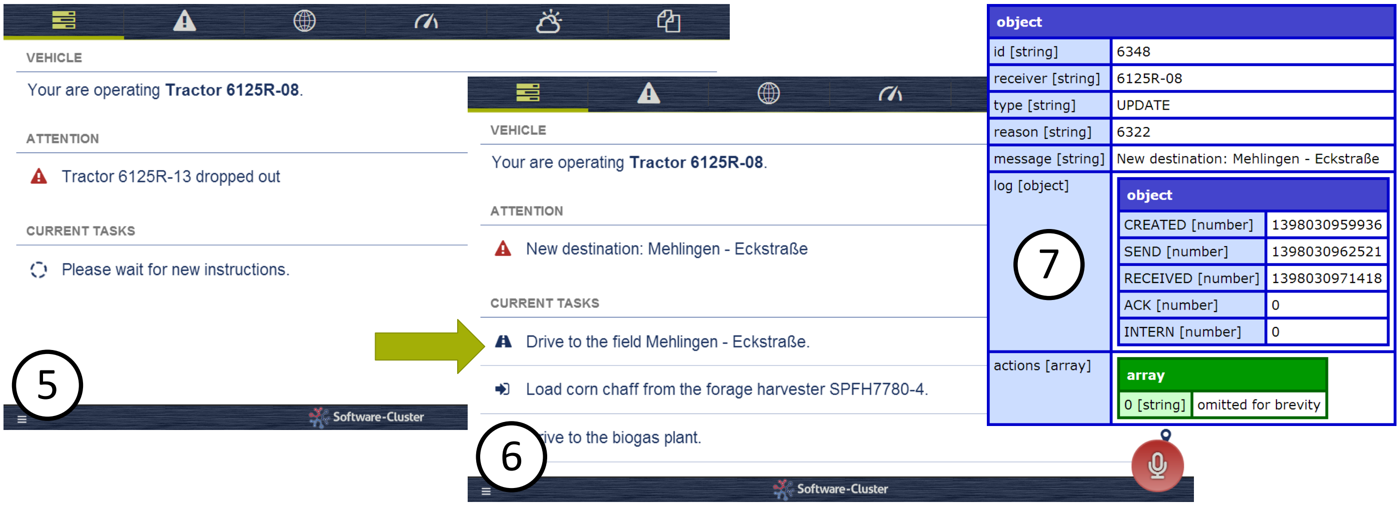

Figure 3. Screen (5) immediately informs an affected driver of an unexpected incident, here the drop out of another tractor. Screen (6) displays an updated list of tasks as reaction to the incident. Screen (7) shows the actual message behind screen (6) in a beautified JavaScript Object Notation (JSON).

was created, when it was sent out, and when it was received and attach this information to the message. This is shown in Figure 3, screen (7). This way, delivery delays, e.g., due to bad network coverage can be detected and separately handled as long as the retransmission approach of messages included in the SIP protocol does not succeed after a specified time. Also shown is that the message is not yet acknowledged and understood by the receiver although it was already successfully delivered. The missing timestamps are added to the message log by the MMDS if the driver responded (ACK) and the MMDS is finally convinced that the driver understood or better internalized the contents of the message (INTERN).

We lend the term *introspection* from psychology as the ability of self-reflection and combine it with the term *task* in order to point out that a driver, when assigned with a new task, first has to reflect if he correctly understood it. If he then comes to the conclusion that something is unclear, he is encouraged to ask the system and thus *inspect* the details in a multimodal dialogue. Currently, a task consists of a location and an action to perform at the location. The MMDS constantly observes the user interaction. Using its dialogue strategy which implements prototypical grounding mechanisms known from dialogue theory [27], the MMDS is able to conclude based on the drivers interactions whether he already internalized his new task, i.e., his new location and action, or not. If not, the MMDS should give or ask for the so far not grounded information and thus stimulating the introspection process of the driver.

Every vehicle in a harvest campaign is equipped with a mobile COAP-RT client. Bob, a tractor driver, wears a bluetooth headset that is connected with the mobile client of his tractor. So, he can benefit from the built-in noise canceling mechanisms of the headset for an improved speech recognition performance and is not required to touch the mobile device while driving. He just unloaded corn chaff at a biogas plant and wants to return to his actual field of activity when an unexpected incident occurs at a different location that affects him. In the following, two alternative examples of typical dialogues between the driver (U) and the system (S) are presented. GUI updates are explained in square brackets.

(1) S: Hello Bob! You will get new instructions due to an unexpected incident. I'll get back with you soon. [*The task view is updated (Figure 3, screen (5)) and the incident appears in the incident view (Figure 2, screen (3), first list item)*]
(2) U: Ok, why that?
(3) S: Tractor 13 operated by Jim dropped out due to a tire damage.
[*The system performs a replanning. The contractor finally agrees to the proposed replanned harvesting process and broadcasts new instructions.*]
(4) S: Attention Bob! You have new instructions. Please drive immediately to the field Mehlingen. [*The task view is again updated (Figure 3, screen (6)).*]
(5) U: Ok, what should I do at the field in Mehlingen?
(6) S: Please support the forage harvester 4 there.
(7) U: Ok, I'm on my way.
(8) S: [*The navigation system is silently updated.*]

In utterance (2), Bob acknowledges receipt of the message with "Ok" and indicates that he understood an incident has occurred with the subsequent "why" question. The answer is given by the system in (3). After replanning, Bob is initially informed about his new task in (4). In (5), Bob again acknowledges receipt of the message. He also repeats his new destination. This immediately indicates the MMDS that he understood the location part of his new task. Since he also actively asks for more information, the MMDS finally assumes in (8) that he also understood the action part his new task. As an alternative to utterance (5), the driver might be less communicative, perhaps due to a higher cognitive load. This results in the following slightly different dialogue.

(1) S: Attention Bob! You have new instructions. Please...
(2) U: Ok, I'm on my way.
(3) S: Your navigation system is updated with Mehlingen as your new destination. Please follow the instructions.
(4) U: Ok, thanks.
(5) S: Please support the forage harvester 4 there.
(6) U: Ok.
(7) U: If something is unclear, don't hesitate to ask.

 ISBN: 978-1-61208-353-7

In (2), Bob acknowledges receipt of the message. He also indicates that he is assigned to a new destination. However, the MMDS is not convinced yet that Bob correctly understood his new destination. Therefore, the MMDS gives an additional hint and repeats the destination in (3). Still, the MMDS is not sure whether Bob knows what to do at his new destination. So, it introduces this information in an anticipatory manner in (5) which is acknowledged in (6). In (7), the MMDS is sufficiently convinced to end the conversation. However, it encourages Bob to ask if questions arise. Please note that the system takes a stronger initiative here in order to achieve a sufficient confidence that Bob understood his new instructions, whereas Bob's utterances are shorter (basically, these are only confirmations) causing less additional cognitive load. Thus, the implemented dialogue strategy is also beneficial for the *safeness* of the driver while driving and inspecting the details of his recently updated instructions.

V. Conclusion and Future Work

We presented the cloud-based COAP-RT system for orchestrating and coordinating a fleet of agricultural machinery and their drivers during an ongoing harvest campaign in case of an unexpected incident. It has been shown how the system contributes to a quick, traceable and safe communication of changed action plans from the contractor to affected drivers. The overall system was successfully demonstrated at the CeBIT 2014 where we got in contact with domain experts who appreciated our approach. Currently, we simulate vehicle drop outs. So, next steps are to get access to real telemetry data, e.g., from a CAN bus or ISOBUS, and to increase the planning granularity in order to be able to conduct a field test for evaluation purposes. Regarding the communication aspect, refinements of the dialogue strategy and related artifacts will be beneficial for a more natural user experience. A user study conducted as an extended lane change test in an appropriate simulation environment as described in [28] can finally asses the driver's cognitive workload. We should also consider more fine-grained escalation mechanisms, i.e., how and when to indicate the contractor that his manual intervention is required.

Acknowledgment

The work presented in this paper was funded by the German Federal Ministry of Education and Research (BMBF) in the Software-Cluster project SINNODIUM (www.software-cluster.org) under grant number "01IC12S01D". The authors assume responsibility for the content.

References

[1] P. Bastian, K. Dagmar, D. Tanja, and S. Albrecht, "Reducing non-primary task distraction in cars through multi-modal interaction," it - Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik, vol. 54, no. 4, Aug. 2012, pp. 179–187.

[2] R. J. Lehmann, R. Reiche, and G. Schiefer, "Review: Future internet and the agri-food sector: State-of-the-art in literature and research," Comput. Electron. Agric., vol. 89, 2012, pp. 158–174.

[3] A. Kaloxylos et al, "The use of future internet technologies in the agriculture and foodsectors: Integrating the supply chain," Procedia Technology, vol. 8, 2013, pp. 51 – 60.

[4] "The iGreen Project," Available: http://www.igreen-projekt.de/ [retrieved: Aug, 2014].

[5] "The iGreen Project - Data Services," Available: http://data.igreen-services.com/ [retrieved: Aug, 2014].

[6] "argoXML," Available: http://www.agroxml.de/ [retrieved: Aug, 2014].

[7] "Linked Data," Available: http://linkeddata.org/ [retrieved: Aug, 2014].

[8] R. T. Fielding, "REST: Architectural styles and the design of network-based software architectures," Doctoral dissertation, University of California, Irvine, 2000.

[9] D. Martini, M. Schmitz, R. Kullick, and M. Kunisch, "Fitting information systems to the requirements of agricultural processes: A flexible approach using agroxml and linked data technologies," in Proc. of the Int. Conf. on Agricultural Engineering (AgEng 2010), 2010, pp. 157–163.

[10] A. Gerber and M. Klusch, "AGRICOLA - Agenten für mobile Planungsdienste in der Landwirtschaft (in English: AGRICOLA - Agents for Mobile Planning Services in Agriculture)," KI, vol. 18, no. 1, 2004, pp. 38–42.

[11] M. Klusch and A. Gerber, "Dynamic coalition formation among rational agents," IEEE Intelligent Systems, vol. 17, no. 3, May 2002, pp. 42–47.

[12] M. Reinecke, H.-P. Grothaus, G. Hembach, S. Scheuren, and R. Hartanto, "Dynamic and distributed infield-planning system for harvesting," in Proc. of American Society of Agricultural and Biological Engineers Annual International Meeting (ASABE-2013), vol. 1. Curran Associates, Inc., 2013, pp. 156–160.

[13] "Farmpilot," Available: http://www.farmpilot.de/com/ [retrieved: Aug, 2014].

[14] "356FarmNet," Available: http://www.365farmnet.com/en/ [retrieved: Aug, 2014].

[15] "Farming 4.0," Available: http://www.laboratories.telekom.com/public/english/innovation/exponate/pages/industrie-4.0.aspx [retrieved: Aug, 2014].

[16] "SAP HANA Cloud Platform," Available: http://www.sap.com/pc/tech/cloud/software/hana-cloud-platform-as-a-service/index.html [retrieved: Aug, 2014].

[17] "Bing Traffic API," Available: http://msdn.microsoft.com/en-us/library/hh441725.aspx [retrieved: Aug, 2014].

[18] "OpenWeatherMap," Available: http://openweathermap.org/ [retrieved: Aug, 2014].

[19] D. Sonntag, D. Porta, and J. Setz, "HTTP/REST-based meta web services in mobile application frameworks," in Proc. of the 4nd Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '10). IARIA, 2010, pp. 170–175.

[20] D. Porta, "A novel, community-enabled mobile information system for hikers," in Proc. of the 2nd Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM '08). IEEE Computer Society, 2008, pp. 438–444.

[21] "SIP: Session Initiation Protocol," Available: http://tools.ietf.org/html/rfc3261 [retrieved: Aug, 2014].

[22] "Session Initiation Protocol (SIP) Extension for Instant Messaging," Available: http://tools.ietf.org/html/rfc3428 [retrieved: Aug, 2014].

[23] R. Neßelrath and D. Porta, "Rapid development of multimodal dialogue applications with semantic models," in Proc. of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems (KRPDS '11), 2011, pp. 37–47.

[24] R. Neßelrath and M. Feld, "SiAM-dp: A platform for the model-based development of context-aware multimodal dialogue applications," in Proc. of the 10th Int. Conf. on Intelligent Environments (IE '14). IEEE Computer Society, 2014, in press.

[25] "Eclipse Modeling Framework Project (EMF)," Available: http://www.eclipse.org/modeling/emf/ [retrieved: Aug, 2014].

[26] "Media Resource Control Protocol Version 2 (MRCPv2)," Available: http://tools.ietf.org/html/rfc6787 [retrieved: Aug, 2014].

[27] H. H. Clark and E. F. Schaefer, "Contributing to discourse," Cognitive Science, vol. 13, no. 2, 1989, pp. 259–294.

[28] A. Castronovo, M. Feld, M. Moniri, and R. Math, "The ConTRe (continuous tracking and reaction) task: A flexible approach for assessing driver cognitive workload with high sensitivity," in Adjunct Proc. of the 4th Int. Conf. on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '12). ACM, 2012, pp. 88–91.