IEEE Software: Special Issue on Observability and Explainability for Software Systems Decision Making

Explainability With Observation Sharing in Long Collaboration Chains of Automated Systems-of-Systems

Paulius Daubaris, *University of Helsinki, Finland*
Simo Linkola, *GIM Robotics* and *University of Helsinki, Finland*
Juan F. Inglés-Romero, *Catholic University of Murcia, Spain*
Javier Berrocal, *University of Extremadura, Spain*
Cristina Vicente-Chicote, *University of Extremadura, Spain*
Juan Manuel Murillo, *Computing and Advanced Technologies Foundation of Extremadura, Spain*
Tommi Mikkonen, *University of Jyväskylä, Finland*
Niko Mäkitalo, *University of Jyväskylä and University Consortium Chydenius, Finland*

Abstract—Robotics enabled by AI and advanced software technologies are taking the world by storm. The advanced automation systems, systems-of-systems that power them, and their collaboration are complex and difficult to understand. Lacking knowledge about how they work can only improve our lives so much; in fact, it can hamper the interactions between systems, harm businesses, disrupt daily lives, and even endanger us. Increasing the observability and explainability of these systems helps us remain in control and leverage the technologies to improve our lives. This article introduces a blueprint architecture for long collaboration chains formed by systems-of-systems, provided by multiple companies, that can significantly improve the understanding of the processes and events taking place. We illustrate the vision with an automated supply chain scenario and utilizing example use cases of how different stakeholders can benefit from the enhanced observability and explainability within long collaboration chains.

Society is becoming ever more automated with the emerging robotic and AI technologies. With the increasing complexity of machine learning and quantity of the produced data, the economies of autonomous systems are in danger of becoming black boxes. Such circumstances can lead to misunderstandings with respect to different system processes involved. This may lead to a lack of accountability, hinder business collaboration and, in the worst case, cause serious harm to humans and their safety (D'Acquisto, 2020). Therefore, it is crucial that humans remain in control and are fully able to understand and make decisions how complex automated systems work (D'Acquisto, 2020). Strengthening the observability and explainability of systems-of-systems (SoS) can help make this reality.

Observability is usually addressed from the operating context and the system maintenance point of view. It is a property of the system capable of retrieving information about the system's behavior and to correlate the behavior of the different modules. Observability allows system maintainers to obtain information about *what*, *how* and *why* something happened on a technical level (Niedermaier et al., 2019; Usman et al., 2022). This information may be used to identify the root

XXXX-XXX © 2024 IEEE
Digital Object Identifier 10.1109/XXX.0000.0000000

This article has been accepted for publication in IEEE Software. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MS.2023.3320742

causes of operational failures. In multi-robot systems, observability can enhance, for example, fault detection and correlation (Khalastchi and Kalech, 2019) across the whole robot population.

Explainability has a similar goal but, instead of focusing on the ability to obtain information about the behavior or the system, it focuses on processing the observed information for the recipient. For instance, in a multi-robot system, the information to transmit a concrete problem about a delayed package should be different for the customer than for the system maintainer. Unlike observability, explainability usually assumes spoken or written natural language output. Explainability is both demanded by governing bodies (Wachter et al., 2017) and a recognized challenge in robotics, with no solution tackling all its issues even within a single robot (Anjomshoae et al., 2019; Sakai and Nagai, 2022).

There are multiple angles from which observability and explainability of a system, and the decision making with the system can be considered and improved. Many of them are subject to who exactly is observing and what needs to be explained. For example, on the one hand, a customer buying a product may be willing to understand the origin of the goods that they have acquired as well as the manufacturing and transportation processes. On the other hand, companies developing autonomous systems desire information about how their systems and software work in production and in collaboration with other companies' systems. For instance, another robot in the logistics chain may need observations, or information about specific characteristics of the product or the logistics that may affect a proper delivery.

At present, many obstacles to improving the explainability aspects of automated, complex systems-of-systems exist. For example, relevant information is often incomplete, lacking context, scattered across various entities of the system, or known by only a few people. Hence, the information is impossible to leverage for understanding and improving SoS.

In this article, we identify the four crucial observability and explainability challenges of automated, complex SoS. We introduce our novel vision of long collaboration chains and a blueprint architecture aiming to resolve the challenges. To highlight the benefits of long collaboration chains, we present an Industry 4.0 scenario of heavily automated supply chains leveraging autonomous robots capable of collaborating with different entities.

Toward Explainability with Long Collaboration Chains

Both observability and explainability make software easier to manage and help in understanding why it made certain decisions. However, catering for both in SoS where different companies, robotic and software solutions, and human partners participate brings specific challenges, especially considering data gathering and sharing.

Observability and explainability require data from the context and the internal state of systems. The same data gathering and analysis processes that are used for observability can be used as the basis for explainability. However, fusing and analyzing the data into meaningful *observations*, i.e., pieces of information considering particular events or phenomena, are not trivial considering the challenges that the physical world imposes on data validity and the data gathering processes.

In ubiquitous machine collaboration, a failure can be a result of multiple incidents that befall to multiple entities and stakeholders, and the first event leading to the failure may have happened a substantial time ago. Therefore, systems need to *share observations* between collaborators to enable observability that transcends the individual company's systems and provides explainability covering the whole collaboration.

Systems need to be able to decide which observations to share and with whom based on the validity of the observations (e.g., trust in the original data source, confidence in the observation correctness) and the collaboration context. Moreover, the solutions that enable observation sharing need to account for, to the best of their ability, disruptions and gaps in the sharing process. For example, some collaborators can be humans who need digital representatives to share the observations with. *What observations to share and with whom to enable observability and explainability are currently an unresolved issue in the context of automated, complex SoS.*

Inspired by Sakai and Nagai (2022) and motivated by the above discussion, we summarize four crucial observability and explainability related challenges of SoS as follows:

Challenge 1: *SoS cannot unravel decision making and overall behavior for humans.*

Challenge 2: *SoS cannot communicate how a single entity's decisions and behavior are estimated to affect the overall decision making and behavior.*

Challenge 3: *SoS do not support extracting all the relevant information from their systems.*

Challenge 4: *SoS cannot communicate explana-*

This article has been accepted for publication in IEEE Software. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MS.2023.3320742

tory factors among their systems.

Our objective is to address these four challenges with a novel *long collaboration chain vision and blueprint architecture*. Our proposed solution focuses on observation sharing during collaboration to ensure that the relevant observations from the context, individual machines and collaboration events are both handed forward in the chain during collaboration and available for later request. To bind systems to share observations, they form *contracts* with one another. A contract obligates the parties to share observations considering relevant viewpoints (e.g., sustainability or safety) of a particular scope, event, or phenomenon (e.g. a single package delivery), enabling the use of various *explanation lenses* – a novel concept that can be applied by different stakeholders during different tasks to better understand SoS behavior and to make wiser decisions.

Scenario: Automated Supply Chains in Industry 4.0

We use an Industry 4.0 supply chain scenario to illustrate how the four previously identified observability and explainability challenges can be mitigated. The scenario as well as the main interactions and solution details are depicted in Figure 1.

The supply chain often involves different stakeholders, including employees who have contrasting responsibilities, multiple vendors (delivering, storing, and manufacturing packages), and end users. Such a setup requires careful thinking about how these parties interact, what information is shared and how their observations are made useful to generate appropriate explanations using particular explanation lenses and in turn improve a company's operations in terms of, for example, sustainability. Figure 1 (I) illustrates five different explanation lenses (A–E), and Table 1 provides related use cases to highlight the benefits of the explanation lenses.

Warehouse AI (Figure 1, center) maintains information about stored packages, deliveries, human and robot employees, and their observations. It collects information to maintain the full picture of the supply chain and to improve the company's business processes (Challenge 3), such as the observations sent by robots. These observations can also be used by collaborating companies and their systems (Company B's Intelligence; Figure 1, top left) to improve their own operations as well. In addition, the accumulated observations are compiled and provided as explanations to hardware and software developers to amend potential system faults and improve the general efficiency of the system (C).

Considering the warehouse environment (E), the warehouse robot leverages sensors and other systems of the warehouse, which are governed by the central warehouse intelligence system. In this case, the robot observes its environment and surroundings and generates an explanation suitable for a human coworker. Such insights aid in maintaining the safety of the warehouse and protecting other coworkers and the robot's surroundings.

While passing a package from the warehouse forward to the delivery (B), two robots from different vendors need to communicate about the package's delivery details. However, since the companies follow strict data privacy rules, exchanging information requires additional safety measures. In case there is no predefined contract between the two companies, one needs to be made on-demand. Therefore, the warehouse robot communicates with the main warehouse intelligence system, sharing its observations about the delivery robot and its company and asking for guidance on how a contract should be made. When the guidance is received, the two robots agree on the terms and observations related to the package that are passed on to the delivery robot (Challenge 4).

The terms agreed on by different companies imply certain obligations. In this particular case, the delivery options and related sustainability aspects are observed(D). Such information can be explained by the robots based on the observations acquired through the entire supply chain to different stakeholders to evaluate the sustainability aspects and methods of delivery (Challenge 2). Such insights can help the user have the power to inspect and choose the most sustainable option.

Upon delivery (A), the delivery robot possesses observations of various steps taken during its operations, which are then used to provide information to the recipient of the package (Challenges 1 and 2). For example, if the recipient receives damaged goods, he or she might demand that the robot explain why the damage happened. This would prompt the delivery robot to reiterate through all the observations that it acquired from the previous robots and to use this information to prepare a suitable explanation taking into account the entity with which it interacts.

Blueprint Architecture for Long Collaboration Chains

Considering the scenario and challenges discussed previously, we propose a blueprint architecture, depicted in Figure 1 (II) which seeks to improve ob-

This article has been accepted for publication in IEEE Software. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MS.2023.3320742

Table 1. Example use cases how different stakeholders can benefit from the different explanation lenses from the observability (O), explainability (E), and decision making (D) perspectives.

Use case		Benefits	Technical Notes
Sustainability Lens **Motivation:** Currently, having a clear and truthful understanding of the sustainability aspects of an automated supply chain is difficult (Challenge 1). For example, information may be stored only in each vendor's own systems, may not be produced or stored at all, or information is not communicated between different systems (Challenges 2 and 3). The sustainability lens provides a tool for a customer to inspect the true carbon oxide footprint and other sustainability aspects of delivery.	**O**	• Based on contracts, different companies' systems are obligated to generate and share observations during the delivery process (Challenges 3 and 4). Other stakeholders can better understand the sustainability viewpoint of the whole process, which thus far has been beyond of their reach (Challenge 1). • For example, entities need to generate observations on CO2 emissions, the packaging materials used during different phases, the origins of the materials, and so forth.	• An SoS entity vendor that has agreed to contract terms, is obligated to make its systems produce and communicate observations. Different software engineering domains use different methods for communicating. For example, in the robotics domain, DDS provides a standardized and efficient data distribution service and allows the creation of a global data space, with access control, security, data persistence, etc., without a central broker (Challenges 3 and 4).
	E	• With the explanation lens, the system is capable of explaining and tracing the different phases and stakeholders involved in the delivery process based on the observations within the supply chain (Challenges 1 and 2). • For example, a stakeholder can query observation data from related systems to understand the sustainability aspects of the entire delivery process.	• Individual observations are stored in the long collaboration chain. Various technologies can be used to store the raw data related to the observations. For example, the InfluxDB database can be connected to DDS to store the entire history of events (Challenge 4). Flux is the advanced query language of InfluxDB for data processing and time series analysis, and it is a good fit to address explainability issues regarding processes that potentially take long periods of time (Challenge 3).
	D	• The stakeholder can select different delivery methods, packaging options or even entire different products that better match the stakeholder's values (Challenges 1 and 2).	• The best configuration could result from an optimization process according to the stakeholder criteria and based on metrics derived from observations (Challenge 2).
Safety Lens **Motivation:** SoS safety becomes a challenge when multivendor systems interact. This is because there are either no observations shared among entities or observations are local to only a single vendor (Challenges 2 and 3). The safety lens provides a tool for observing how various systems interact and then explaining this behavior to humans (Challenge 1). Based on these explanations, humans can better detect potential safety issues, such as how well machines communicate when a human is detected, and how other machines are expected to react to a human's actions (Challenges 1 and 2).	**O**	• Entities make important observations of their own and other entities behavior, location, etc., and then share these with the long collaboration chain (Challenges 3 and 4). • For example, while robots co-operate and interact with each other and other systems in the warehouse, they generate local observations about their own operating environment including the entities surrounding them, and whether they are other machines or human coworkers (Challenges 3 and 4).	• Complex information derived from the context can be shared using different technologies (Challenges 3 and 4). For example, a robot can perform facial recognition of co-workers and update a specific DDS Topic with their identities and locations. Another system can follow this topic and observe that an unauthorized coworker was in a prohibited area dedicated to the transit of goods.
	E	• The system can provide explanation views that help different stakeholders improve safety aspects in different situations (Challenge 1). • For example, based on the various observations made by different robots, stakeholders (the warehouse robot maintainer, robot developers, etc.) can better identify SoS safety issues caused by the interplay of robots when those aspects cannot be guaranteed (Challenges 1 and 2).	• In general, by leveraging CEP technologies, observations can be traced back to the original events that produced the detection of the corresponding pattern (Challenges 2 and 3). Moreover, in robot-to-robot interactions, InfluxDB can be used for making queries to the raw observation data, which help in explaining causes and effects for stakeholders (Challenges 1, 2 and 3).
	D	• Based on the explanations provided by the system, stakeholders can act to improve the safety of the warehouse and its operations (Challenge 1). • For example, the warehouse robot maintainer could improve the network coverage in case the robot has to halt its operations due to poor connectivity, or fix possible issues with the robot hardware or in the robot's physical environment.	• Probabilistic networks can be used to estimate the level of safety of a system based on observations. The resulting metric provides a coarse-grained measure of how well the system performs, which can support the decision-making process (Challenge 1).

This article has been accepted for publication in IEEE Software. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MS.2023.3320742

Use case		Benefits	Technical notes
Optimization Lens **Motivation:** Vendors provide tools for developing and optimizing their system's behavior. However, when deploying multivendor systems, the overall system-of-system observability and explainability are hard to achieve (Challenges 1-4). The optimization lens provides intersystem tool for understanding behavior and optimizing system performance (Challenges 1 and 2).	**O**	• Robot tracks its processes and simultaneously observes current operations and measures their efficiency, and the stability among other aspects (Challenge 3). Anomalies are shared with the long collaboration chain (Challenge 4). • For example, two robots collaborating closely can report their progress as observations and share them with others (Challenges 3 and 4).	• CEP technologies can be used for describing observations as expressions of event patterns, e.g., using the Event Processing Language (EPL) of Esper (https://espertech.com/) (Challenge 4). This approach requires precise pattern definitions, which are not always possible. In this case, fault detection techniques, e.g., based on autoencoders or transformers, can be used to identify deviations from normal behavior (Challenge 3).
	E	• The calculated measurements are streamlined into a dashboard-like explanation view with various insights about the robot's decisions and operations in general that are further processed for easy-to-understand insights, which is useful for the developers (Challenges 1 and 2). • For example, developer can track each step of two robots collaborating and observe why performing certain tasks has taken a considerable amount of time (Challenge 2).	• There are several tools that can be used for visualizing observations which essentially are time-based data. For example, Grafana (https://grafana.com/) is a data visualization tool for creating interactive dashboards for time series data. It can support explainability by visualizing outputs, metrics, and contextual information (Challenge 1).
	D	• Using the measurements and the explanations of robot actions, developers can enhance the system relatively easily, as debugging and tracking down system faults is already alleviated by the insights provided by the robot explanations (Challenges 1 and 2).	• The nonfunctional properties of a system can be observed with runtime quality of service metrics (see Sidebar), and provide support for decision making and improving performance (Challenges 1 and 2).
Privacy Lens **Motivation:** Currently, an ever-increasing number of AI and other software systems produce, store, share, and use the personal information of their users. It is becoming impossible for any individual human to keep track of and understand what sensitive information these numerous systems possesses and how this information is being used (Challenges 1–3). The privacy lens provides a tool for the user to inspect what private information each system of an automated supply chain possesses and how these systems have reported using this information. The privacy lens can also help system vendors support EU's GDPR and AI Act (Challenges 1 and 2).	**O**	• New privacy observations are generated when a robot or any other system stores or accesses user's information (Challenge 1).	• Raw data should be stored in actual entities and systems. The long collaboration chain stores anonymized observations reported by the entities. For example, with proper access control, InfluxDB can provide the necessary querying and analysis tools to facilitate transparency and understanding of data usage in robotics applications (Challenges 3 and 4).
	E	• With an explanation lens, a user can select a scope and study what entities have accessed, stored, or handled private information about the user (Challenge 1). The user can also understand for what purpose the data has been used for (e.g., training a machine learning model) (Challenges 1 and 3).	• By using an ontology to build a taxonomy and link semantics and implications, advanced access control to information can be established based on whether it is subject to legislation, among other aspects (Challenges 1 and 3).
	D	• Users can better understand their data usage and make decisions to ask different stakeholders to delete private information, including privacy observations (Challenge 4). System vendors can also understand how other systems use their produced data (Challenges 1, 2 and 4).	• In private data management, user requests and actions by the corresponding company can be registered with blockchain technology, providing a decentralized and transparent mechanism to ensure data integrity and trust. To omit data from the long collaboration chain, personal data is anonymized using hash values and public key cryptography (Challenge 4). Correspondingly, system vendors take care of deleting the raw data from their services – as they would normally do. The long collaboration chain keeps track of these changes and privacy lens can then be used to ensure that the data have actually been removed (Challenges 1–4).

This article has been accepted for publication in IEEE Software. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/MS.2023.3320742

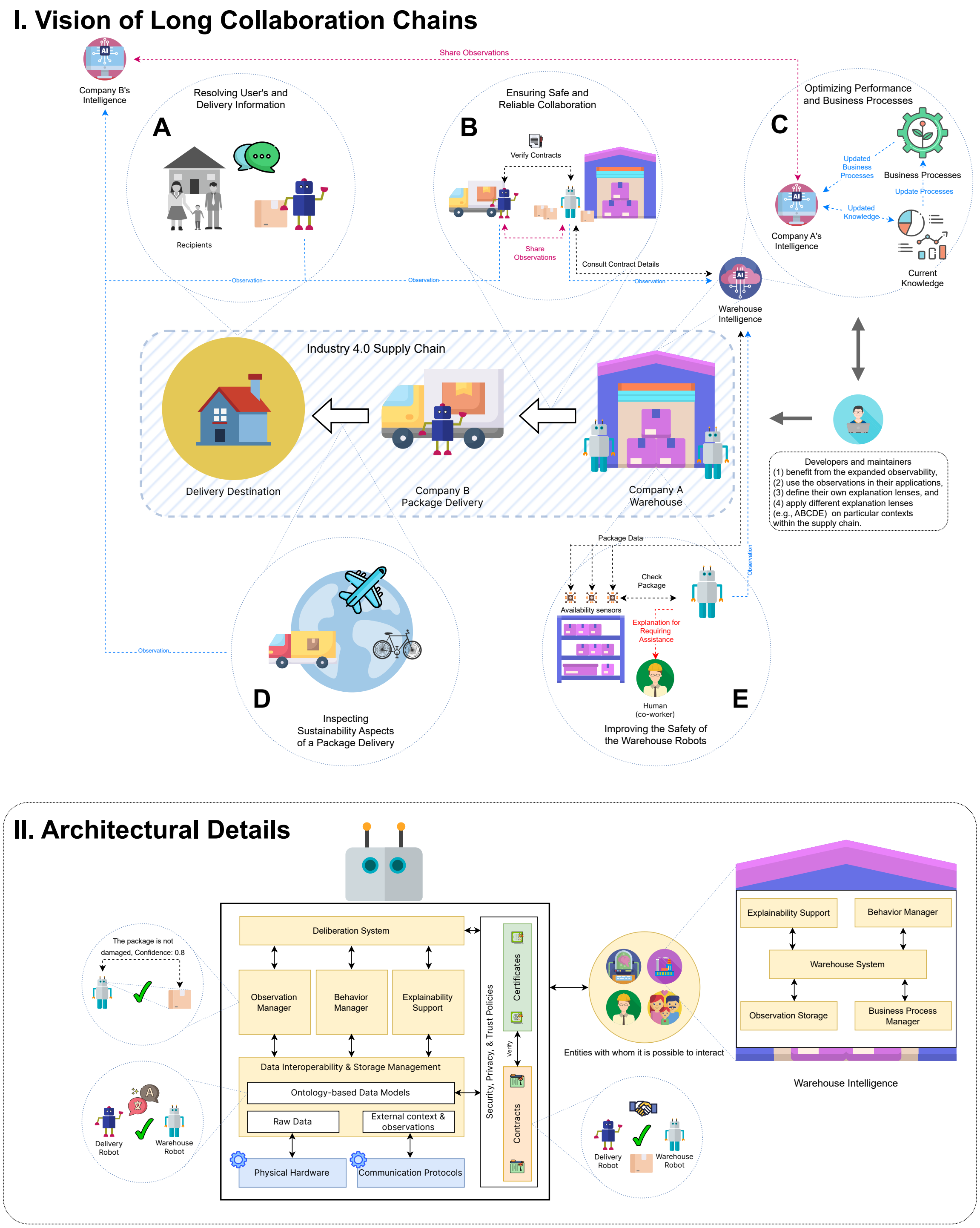

Figure 1. I. An illustration of the long collaboration chain vision with an industry 4.0 supply chain. Sharing observations between individual robots (B) and company intelligences (top) enables more comprehensive observability for system maintainers and developers (right middle) and allows explanations for different stakeholders (A, B, C, D, E, right middle).
II. Blueprint architectures of the robot and the centralized system, in this instance the warehouse intelligence.

Motivation and Technologies for Improving Observability and Explainability

Business Incentives for Multivendor Data Sharing

The autonomous machine industry aims to boost revenue through business collaboration with other hardware and software vendors. Companies need to (a) be able to trust other systems and (b) be able to reliably prove and provide explainability of their own systems operation. A Bad reputation can harm business. Platform Economy for Autonomous Mobile Machines project explores collaborative opportunities in robotics, such as five companies building an automated warehouse system using each other's components.

Research-based Solutions for Improving Observability of Quality-of-Service

Nonfunctional properties like safety impact *how* a system works, not just *what* it does. In the RoQME project, we offered a framework to address these properties with global QoS metrics. Their real-time estimation, based on robot operation observations, aids in behavior adaptation and requirement assessment. We tested the project in an intralogistics setting, involving factory goods transport (Vicente-Chicote et al., 2019).

Robot-to-Robot Collaboration and System Interoperability

Robot-to-robot collaboration is challenging, particularly in multi-vendor environments where different technology stacks are involved. In our project on Creative and Adaptive Cooperation between Diverse Autonomous Robots, we've explored the use of ontology-based communication methods to facilitate information sharing and physical coordination in diverse robot populations (Linkola et al., 2022).

References

Linkola, S., Mäkitalo, N., Laurinen, T., Kantosalo, A., and Männistö, T. (2022). An architectural approach for enabling and developing cooperative behaviour in diverse autonomous robots. In *Software Architecture: 15th European Conference, ECSA 2021 Tracks and Workshops, Revised Selected Papers*, pages 181–204. Springer.

Vicente-Chicote, C., García-Pérez, D., García-Ojeda, P., Inglés-Romero, J. F., Romero-Garcés, A., and Martínez, J. (2019). Modeling and estimation of non-functional properties: Leveraging the power of QoS metrics. *From Bioinspired Systems and Biomedical Applications to Machine Learning*, pages 380–388.

servability and explainability. We envision two different kinds of systems: robots (or, by extension, autonomous systems), providing local observations and explanations to their collaborators; and warehouses, maintaining integrated views of all the observations and their relations with the business processes being executed.

Robot's Software Architecture

Next, we describe the architecture's main modules of a robot operating in the supply chain.

Physical Hardware. The robot's physical hardware is used to sense the context, and provide other functionalities, e.g., movement through actuators. The context data together with the internal state of the robot, and the actions and interactions performed, are later used to detect events of interest and produce observations from them. Figure 1 illustrates physical hardware as an abstract component. It is abstracted because the physical hardware could comprise different sensors and actuators whose differences from the blueprint architecture point of view do not matter.

Communication Protocols. In multivendor collaborative scenarios (e.g., different robots involved in the supply chain), communication between systems in

This article has been accepted for publication in IEEE Software. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MS.2023.3320742

the environment is vital. Multivendor systems present challenges because of diverse technology stacks and communication standards. In previous projects, we've addressed this by using Data Distribution Service (DDS, https://www.omg.org/spec/DDS) as a common middleware for efficient data distribution across vendors. The data format in DDS is explicitly defined using DDS Interface Description Language (IDL).

Security and Privacy. Data security, privacy and trust are key when information, such as observations, and needs to be shared with different entities. Communications must be secure, and robots should only share essential information to protect company and personal data. This module uses contracts to specify data-sharing conditions.

For example, when moving a package from the warehouse to the delivery truck (Figure 1, B), the warehouse robot has to collaborate with the delivery robot. If these robots belong to different companies, it will be necessary to sign a contract before the collaboration (if it does not yet exist), for example, to establish what information can be shared (Challenges 3 and 4). Contracts can be generated on demand for specific tasks, through an automatic process that involves the parties negotiating the terms. This process can follow a similar approach to QoS (quality of service) negotiation in service composition (Buscemi and Montanari, 2011). The resulting contract is signed with a certificate-based signature. Note that the contract format should be designed to be readable by both humans (since it is a legal document that binds the two companies) and machines (which must manage the interaction within the agreed-upon limits).

Although the notion of contract is not inherently part of DDS, it could be implemented at the application layer. Additionally, DDS Security (https://www.omg.org/spec/DDS-SECURITY) was released as an extension to the standard to enable advanced authentication, access control, and encryption, among other aspects, providing the basis to develop the security and privacy module.

Data Interoperability and Storage Management. While the communication layer may predefine the format of the data exchanged (e.g., using IDL in DDS), the meaning of these data could still be subject to misinterpretation by the parties involved. As a result, this module is responsible for providing common semantics associated with context data, metrics, actions, and observations to enable interoperability (Challenge 1). In this sense, an ontology-based approach can help consolidate the information among systems. In addition, this module stores the history of events (including context and metric updates, actions taken, and observations) to allow history-based explanations (Challenge 2).

Deliberation System. This module enables the base operations of the robot, including monitoring, observing, acting, planning, learning, and goal reasoning (Ingrand and Ghallab, 2017), among others. The dynamic selection of an appropriate behavior implies establishing the causal relationships that enable the prediction of how certain changes in the robot's operation will impact its performance (Challenge 2). The simplest approach is to establish a set of adaptation rules at the design time. More advanced techniques include reinforcement learning and multicriteria optimization.

Observation Manager. Robot-monitored data or information from other systems can be processed to derive new observations and obtain metrics on system performance, QoS, or task completion. These metrics must align with company objectives. Given the varied sources of observations (e.g., other robots or diverse data types), they might have different confidence levels. For instance, the system might report "the package is undamaged" with 80% confidence due to potential inaccuracies in information sources.

Observations are event-type patterns predefined based on context variables and detected at runtime using complex event processing (CEP) techniques. We employ the Esper platform (https://www.espertech.com/esper), which enables observations like “the delivery robot collided with a fragile package”. Esper facilitates event pattern identification, considering time and aggregated data.

In our observability model (Romero-Garcés et al., 2020), an observation can indicate system performance regarding non-functional properties. For instance, “robot collision during package delivery” impacts the expectation of successful delivery. Using a probabilistic method, we can estimate this expectation with a performance metric.

Behavior Manager. Observations (and the resulting metrics) provide a means of assessing the robot's behavior on the assigned tasks. Therefore, if the robot observes a decline in performance, this module provides adjustments to the robot to improve its operation.

We use behavior trees for the decision-making processes and the behavioral logic of the robot. By employing behavior trees, we can effectively represent complex behaviors, define the hierarchical structure of actions and conditions, and manage the flow of control in a clear and organized way. This allows us to design adaptable systems that can respond to observations (Romero-Garcés et al., 2022).

Explainability Support. This module provides stakeholders with the tools to facilitate explainability, limited to the information stored in the system. It can be considered part of a larger ecosystem, comprising the systems involved in the supply chain to collectively support explainability through the exchange of context, metrics, actions, and observations.

This module leverages: (1) databases like InfluxDB with Flux for efficient time-series data handling; (2) data visualization tools like Grafana for insights via interactive dashboards; (3) traceability in CEP to track observation sources; and (4) metrics to highlight system facets for improvement. Explanations should be tailored to the audience, ensuring non-technical users aren't overloaded and respecting privacy guidelines from the Security and Privacy module.

Warehouse's Software Architecture

Each robot has a local view of the collaboration process being executed. Therefore, the central warehouse needs to obtain a general view of the whole process to identify deviations and improve the coordination between entities. To that end, we envision a system with, at least, the following modules:

Business Process Manager. It stores the business processes to identify the correctness of the executed actions, the deviations, and the reasons behind each deviation.

Storage Management stores all the information provided by robots to obtain a general view, which can be combined with the business processes to trace back actions and observations (Challenge 2).

Behavior Manager. The obtained general view is used to identify predictions, bottlenecks, etc. In addition, different recommendations can be identified to adapt the behavior of the whole collaboration chain to better meet the business goals (Challenge 2).

Explainability Support. Observations and recommendations should be explained to operators and analysts, integrated with business processes, and presented in business-friendly language, as discussed in Table 1.

Other Challenges

Our solution does not take a stance on all the challenges of observability and explainability in automated, complex SoS. Below, we point out a few fundamental challenges.

Observation uncertainty and communication. Local observations made with physical sensors of the real world are noisy and uncertain. To enable reliable communication of these observations across machines, the first step is for the machines to analyze and validate their own observations. However, as the observations are shared with other machines, we also need to be sure that the machines do not propagate errors in the communication chain by misinterpreting the observations, which require unified standards for how the observation scope, content, and validity (e.g., confidence) are inferred and communicated.

Aligning observations and fault detection. Collaborating robots should agree on the observations that they make about the collaboration. While different agreement algorithms exist, it is not currently clear how close to each other the observations should be for agreement. Furthermore, if robots disagree, which robot(s) should be believed? Broken physical hardware can cause incorrect assessments of the collaboration, and should be taken into account in the reasoning for increased reliability.

Trust in observations and explanations. Trust is a social property that is separate from observation or the explanation confidence computations of single entities. A robot with a low-fidelity sensor may have high confidence in its own observations while others assess them incorrectly. Thus, long collaboration chains need shared trust mechanisms that take into account the hardware and software limitations – as well as malicious intents – of its actors so that only appropriately trustworthy observations and explanations are propagated along the chain.

Business incentives for long collaboration chains. In multivendor autonomous systems, software companies must trust each other and demonstrate their solutions' reliability (see Sidebar). Legislation like the EU's explainable AI and GDPR make companies accountable. Our solution improves collaboration by breaking down data "silos" and allowing inspection of data handling between entities.

Conclusions

Considering the advancements of robotics and artificial intelligence, as well as the improved interoperability between systems, we will witness ever more automated and complex software systems in the coming years. To leverage these systems efficiently and effectively, all parties involved (users, system administrators, other systems, etc.) must be able to understand how each subsystem works. This can be achieved only if observations can be obtained, shared and appropriately explained to other systems and stakeholders.

In this paper, we identified four crucial challenges that may affect how well automated, complex SoS can serve humans. To resolve these challenges, we presented a vision of long collaboration chains and illustrated the vision in the context of an Industry 4.0

supply chain. We also provided a blueprint architecture that seeks to address the identified challenges by leveraging the described components and realizing the vision. With these suggestions, we anticipate that intelligent systems will become more observable and explainable, thus enabling different stakeholders to make better decisions and leverage technological advancements in more sustainable ways.

Acknowledgments

This work was supported by Business Finland and Academy of Finland project #328729.

References

Anjomshoae, S., Najjar, A., Calvaresi, D., and Främling, K. (2019). Explainable agents and robots: Results from a systematic literature review. In *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems.

Buscemi, M. G. and Montanari, U. (2011). Qos negotiation in service composition. *The Journal of Logic and Algebraic Programming*, 80:13–24.

D'Acquisto, G. (2020). On conflicts between ethical and logical principles in artificial intelligence. *AI & SOCIETY*, 35(4):895–900.

Ingrand, F. and Ghallab, M. (2017). Deliberation for autonomous robots: A survey. *Artificial Intelligence*, 247:10–44. Special Issue on AI and Robotics.

Khalastchi, E. and Kalech, M. (2019). Fault detection and diagnosis in multi-robot systems: A survey. *Sensors*, 19(18).

Niedermaier, S., Koetter, F., Freymann, A., and Wagner, S. (2019). On observability and monitoring of distributed systems–an industry interview study. In *Service-Oriented Computing: 17th International Conference, ICSOC 2019, Toulouse, France, October 28–31, 2019, Proceedings 17*, pages 36–52. Springer.

Romero-Garcés, A., Freitas, R. S. D., Marfil, R., Vicente-Chicote, C., Martínez, J., Inglés-Romero, J. F., and Bandera, A. (2022). Qos metrics-in-the-loop for endowing runtime self-adaptation to robotic software architectures. *Multimedia Tools and Applications*, 81:3603–3628.

Romero-Garcés, A., Martínez-Cruz, J., Inglés-Romero, J., Vicente-Chicote, C., Marfil, R., and Bandera, A. (2020). Measuring quality of service in a robotized comprehensive geriatric assessment scenario. *Applied Sciences*, 10.

Sakai, T. and Nagai, T. (2022). Explainable autonomous robots: a survey and perspective. *Advanced Robotics*, 36(5-6):219–238.

Usman, M., Ferlin, S., Brunstrom, A., and Taheri, J. (2022). A survey on observability of distributed edge & container-based microservices. *IEEE Access*, 10:86904–86919.

Wachter, S., Mittelstadt, B., and Floridi, L. (2017). Transparent, explainable, and accountable ai for robotics. *Science Robotics*, 2(6):eaan6080.

Paulius Daubaris is a doctoral researcher at the Department of Computer Science of the University of Helsinki, where he also finished his M.Sc. in 2021. His research interests involve software engineering for robotic systems, machine learning for resource constrained devices, edge computing and IoT. Contact him at paulius.daubaris@helsinki.fi.

Simo Linkola finished his Ph.D. in 2022 at the Department of Computer Science of the University of Helsinki where he worked as a postdoctoral researcher until June 2023. His research revolves around autonomous agents and robots, with special interest in collaboration and creativity in machines. Currently he works at GIM Robotics. Contact him at simo.linkola@gimrobotics.fi.

Juan F. Inglés-Romero is a postdoctoral researcher at the Catholic University of Murcia (Spain). Contact him at jfingles@ucam.edu.

Javier Berrocal (Member, IEEE) is an associate professor at the University of Extremadura. His main research interests are software architectures, mobile computing, computing continuum and digital twins. Contact him at jberolm@unex.es.

Cristina Vicente-Chicote is currently an associate professor with the Department of Informatics and Telematics System Engineering, University of Extremadura, Badajoz, Spain. Contact her at cristinav@unex.es.

Juan Manuel Murillo (Member, IEEE) is a full professor at the University of Extremadura and the general manager of the Computing and Advanced Technologies Foundation of Extremadura. His main research interests are software architectures, mobile computing and Quantum computing. Contact him at juanmamu@unex.es.

Tommi Mikkonen is a full professor of software engineering at the University of Jyväskylä, Finland. He received his Ph.D. in Software Engineering from Tampere University of Technology, Finland year 1999. His current research interests include IoT, software engineering for robotics, and multi-device programming. Contact him at tommi.j.mikkonen@jyu.fi.

Niko Mäkitalo is an assistant professor at the University of Jyväskylä, Finland. Niko shares his time with the Faculty of Information Technology and the University Consortium Chydenius. He received Ph.D. degree in computer science from the Tampere University of Technology, in 2016. His research interests are robotics, system architectures, IoT and Fog/Edge Computing. Contact him at niko.k.makitalo@jyu.fi.