ITU Journal on Future and Evolving Technologies, Volume 6, Issue 3, September 2025

Article

STRAUSS: Scalable intent-driven industrial network service quality assurance with asset administration shells

Deniz Cokuslu[1], Ajay Kattepur[2]

[1] Ericsson Research Networks, Turkey, [2] Ericsson Research Artificial Intelligence, India

Corresponding author: Deniz Cokuslu, deniz.cokuslu@ericsson.com

With the emergence of 6G, there is an increasing need for autonomous network service quality assurance. This process is particularly challenging at scale when: (i) requirements of tens of industrial components are to be considered; (ii) conflicting requirements are to be resolved; (iii) mapping needs to be done across industrial and 6G domains; and (iv) autonomous intent resolution is to be incorporated. In this paper, we propose STRAUSS, a scalable intent-driven framework for service quality assurance. We exploit the features of asset administration shells to effectively manage requirements towards the network in an interoperable manner. The intents are mapped effectively between domains using AI-driven techniques. A similar Artificial Intelligence (AI)-driven technique is used to decompose intents with end-to-end expectations to various resource level domains to enable autonomous network configuration management. The proposed system is demonstrated over a realistic use case with multiple picking robots and Autonomous Mobile Robots (AMRs).

Keywords: AI, asset administration shell, intent, interoperability, scalability, service quality assurance

1. INTRODUCTION

The next generation cellular network evolution towards 6G has brought in new use cases related to digital twins, extended Reality (XR) and autonomous vehicles [1]. Of particular interest, are the Industry 4.0 use cases that allow autonomous mobile robots to autonomously operate by making use of 6G-enabled obstacle avoidance, target detection and localization.

The evolution of 6G has enhanced the need for autonomous management of networks due to increased complexity and scale [2]. To this end, one promising direction is the use of intent-driven operations [3], that provide a declarative technique for the management of networks. Intent-based management principles, when combined with Artificial intelligent (AI) agents, enable networks to be configured autonomously to meet particular targets, goals or expectations under given constraints. It is also a promising solution to enhance the degree of automation to achieve digitalization of the enterprises in an end-to-end manner. However, while there are quite a few papers targeting intents over public networks, extending the application of intent-based automation to address industrial requirements in private network deployments is not straightforward.

One of the key requirements to achieve holistic a view of automation in private networks is interoperable operations and orchestration across domains. In this direction, one of the well-known and widely adopted solutions in industrial domains is Asset Administration Shell (AAS) [4]. AAS serves as a digital twin [5] for robots, devices and production lines on the factory floor. The advantage of AAS is the vendor-agnostic coordination that is possible amongst multiple components from Industry 4.0. As an extension, 5G-AAS has been proposed [6], that can allow seamless coordination between industrial and 6G network components.

To effectively integrate 6G into enterprise automation, such as smart manufacturing and automated warehouses, there is a need for an intelligent solution to ensure network Service Quality Assurance (SQA) [7]. SQA techniques provide capabilities that effectively map the requirements from the enterprise domain to the 6G domain and implement optimal configuration. These assurance systems must also reconfigure autonomously with new intents or traffic pattern changes. In our previous studies, we have looked at integrating SQA with industrial assets [8]. However, there were limitations in the approaches:

- The number of robots were assumed to be limited due to the AAS implementation.
- Intent managers cannot process excessive number of intents that are received simultaneously when scale increases.
- Conflicts in intents could not be resolved effectively.
- AI algorithms were used in a limited manner to map requirements from the industrial domain to the 6G domain.

To meet the above challenges, we propose STRAUSS, a scalable intent-driven framework for network SQA. STRAUSS uses advanced features of AAS for enabling interoperability between domains and grouping requirements to services. This process, in turn, improves the conflict handling functions and scalability of the overall intent management system. Furthermore, multiple levels of AI-based requirement mapping and hierarchical intent management are used to further improve the scalability of the proposed approach. STRAUSS is demonstrated over a realistic Industry 4.0 use case involving multiple robots.

The principal contributions of this paper include:

1. Efficient grouping of intent requests within the AAS and eliminating submission of redundant intents.
2. Mapping of requirements using AI techniques.
3. Hierarchical management of intents.
4. Autonomous handling of trade-offs between performance and energy efficiency.

The rest of the paper is organized as follows: Section 2 presents background information on scalability, intent-driven networks, asset administration shells and POMDPs. Section 3 presents the STRAUSS framework with details on the efficient intent grouping, QoS mapping and hierarchical intent decomposition techniques. Theoretical analysis of scalability is presented in Section 4. Section 5 evaluates the STRAUSS framework against alternatives to demonstrate scalability. Related work is presented in Section 6. Section 7 provides conclusions and future directions.

2. BACKGROUND

The main objective of this paper is to address the scalability issues in the intent-based network SQA framework, in which AAS and AI-driven mapping techniques play an important role. Hence, first it is needed to understand the exact characteristics of the scalability issues in such environment, along with the technology capabilities.

2.1 Scalability

Scalability is the ability of a technical system to handle an increasing workload without a corresponding increase in the complexity or cost of the system. In other words, a scalable system can adapt itself to manage growth in data or workload while maintaining a certain level of performance. A system is considered as scalable under these following conditions:

- Add resources easily when demand or workload increases.
- Remove resource easily when demand or workload decreases.

The ways of measuring the degree of scalability can vary, depending on the aspect of the evaluation [9]. Different scalability measures and dimensions exist in the literature. One of them is administrative scalability, which addresses the increasing number of organizations or users to access a system. It is associated with the ability to on-board multiple device types. On the other hand, functional scalability is related to the integration of a new functionality without disrupting existing activities. These definitions, along with another set of scalability aspects, are also summarized in Table 1.

2.2 Intent-driven networks

With the increased complexity of the network and the integration of value added services on top of the connectivity, there is a need to enhance the degree of automation in the network management. Leveraging AI techniques, intent-based management is one of the key enablers for achieving the desired level of automation in the network. Intent-driven networks refer to a system model with automated execution by integrating additional functionalities on top of the policy-driven operations.

According to the definition by TM Forum [3], "intent is formal specification of all expectations including requirements, goals and constraints given to a technical system". Based on this principle, Intent Management Function (IMF) instances at different layers of network management stack automatically defines policies for optimal network management. Given an intent with various

Table 1 – Scalability definitions and aspects

Scalability	Definition
Administrative	Supporting increasing number of organizations or users to access a system
Functional	Integration of a new functionality without disrupting the existing activities
Geographic	Expansion from a local to larger area
Load	Expand/contract to accommodate heavier/lighter load in the system
Generation	Adopting new generation of a component
Heterogeneous	Adopting components from a different vendor

expectations, IMFs monitor the network and collect measurements. If there is a violation of any expectation, IMF starts generating potential action proposals. Based on the predicted impact on the system state, each action proposal is evaluated by IMF. By taking the utility of each intent into consideration, IMF tries to pick the action that is expected to maximize the global utility of the system.

Not just the complexity of the next generation cellular networks, but also requirement of enhanced scalability from different aspects render intent-based management as one of the key pillars of the network service quality assurance framework. With minimum human intervention, the network should scale itself accordingly for increased or decreased load. On one hand, this requires the network to adapt itself to dynamic conditions. On the other hand, these processes should be executed automatically for achieving high performance and efficient reconfiguration of the network. Based on the intent oriented operations and the concept of utility, intent-driven networks pave the road towards higher level of autonomy.

As mentioned, for efficient handling of intent lifecycles at various layers, TM Forum also introduces the design of IMF hierarchy [10]. From business to the resource layer, we envision an intent-based management framework with a holistic automation view. This design choice increases scalability, as well as the resolution of management schemes at different layers and domains. Upon receiving an intent focusing on business objectives, the IMF hierarchy breaks down the expectations and goals into resource layer intents to be handled in separate network domains (e.g., Radio Access Network (RAN), core network).

In addition to the technical advancements, there is an ongoing effort in standardizing intent modelling, interfaces and other aspects of intent-based automation. There are work groups in different standard development organizations focusing on intents such as TM Forum, ETSI, 3GPP and O-RAN.

2.3 Asset administration shell

Asset Administration Shell (AAS) is proposed by Platform Industrie 4.0 [4] as a framework to create digital twins of the industrial assets. It is one of the key enablers

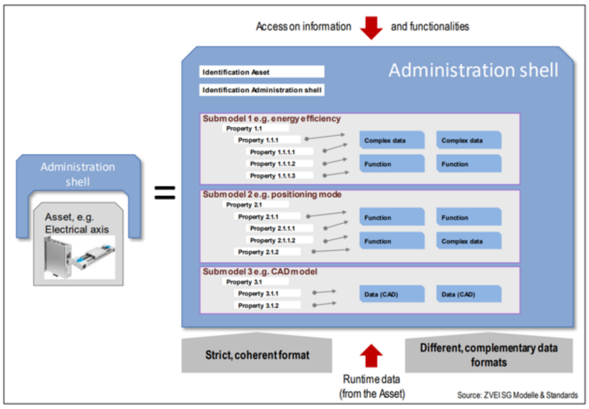

Figure 1 – Asset administration shell example

for the digitalization of the enterprises by integrating the assets into digitalized processes and create a common understanding throughout the Industry 4.0 architecture. AAS consists of data structures called "submodels", which store and provide all relevant information about the asset. Each submodel in AAS represents a particular aspect of the asset. For example, energy efficiency and connectivity capabilities can be designed as separate submodels of a given asset. Based on the information stored in submodels, an AAS may communicate with other AAS instances for enabling collaboration through standardized communication protocols, such as MQTT or REST Application Programming Interfaces (APIs). AAS also enables communication capabilities with I4.0 devices via proposed standards VDI/VDE 2193. An example representation of AAS is shown in Fig. 1.

AAS can be in either passive or active form. If AAS is used as a module representing the capabilities, state and functions of the corresponding asset in the digital world without any interaction capabilities, then it is considered as passive AAS. However, if AAS is integrated with the capabilities of exchanging information and interacting with other AAS instances, it becomes "active". On a factory floor, AAS's for various assets are contained by a central AAS server. The AAS's can be onboarded to this server via either REST APIs, AASX file, or AAS Software Development Kits (SDKs) such as those provided by BaSyx[1]. Active part of the AAS on the other hand, can either be developed using AAS invoke operation, AAS SDKs or a separate application that communicates with the corresponding AAS via AAS exposures. Depending

[1] https://eclipse.dev/basyx/

on how AAS passive and active parts are developed, the resource requirements vary.

Scalability of AAS on a factory floor can be examined in two aspects: (i) communication scalability; (ii) computational scalability. These are highly dependent on how implementation is planned. These aspects are inversely proportional depending on which method is used in implementation. When SDK is used for active parts, computational resource requirements increase due to heavy utilization of BaSyx libraries, but communication requirements decrease. When they are implemented as individual services that communicate with AAS Server via standard exposures, such as REST API, then BaSyx libraries are not used, thus computational requirements decrease and communicational requirements increase.

For the STRAUSS framework, it is assumed that the scale of devices, thus active parts is high. Another assumption is that the framework, including the AAS server is deployed on the edge, on the same cluster with the 6G NW. Therefore, large-scale communication between components is assumed to be handled by the network natively. On the other hand, in a factory environment, where number of device AASs is expected to be hundreds, or thousands, memory limitations arise when BaSyx libraries are used for the active part. Considering these observations, we decided to develop active parts of the device AAS as individual lightweight microservices, communicating to the passive AAS via REST API requests through an AAS server.

2.4 Partially observable markov decision process

In order to efficiently map the states, observations, and actions between the industrial and 6G domains we make use of Partially Observable Markov Decision Processes (POMDPs):

Definition 1 *POMDP* A **Partially Observable Markov Decision Process (POMDP)** is a tuple $\langle S, A, \Omega, T, O, R \rangle$:

- S is a finite set of states.
- A is a finite set of actions.
- Ω is a finite set of observations.
- T is a transition function defined as $T : S \times A \times S \to [0, 1]$.
- O is an observation function defined as $O : S \times A \times \Omega \to [0, 1]$.
- R is a reward function defined as $R : S \times A \times S \to \mathbb{R}$.

With a set of states $S = \{s_1, \ldots, s_N\}$ and a set of actions $A = \{a_1, \ldots, a_K\}$, when the agent takes an action a in state s, the environment transitions to state s' according to the transition function $T(s, a, s')$. The agent receives a reward $R(s, a, s')$ as a result of this action. The set of observations $\Omega = \{o_1, \ldots, o_M\}$ represent all possible sensor readings the agent can receive. The observation is dependent on the function $O : S \times A \times \Omega \to [0, 1]$, with independently tracked probability of observations.

The goal of the agent is to choose actions which fulfil its task as well as possible, thus generating an optimal policy. In POMDPs, a policy $\pi(b)$ maps beliefs to actions over a continuous set of probability distributions over S. A policy π can be characterized by a value function $V^{\pi}(b)$ which is defined as the expected future discounted reward the agent can gather by following π starting from belief b:

$$V^{\pi}(b) = E_{\pi}\left[\sum_{t=0}^{h} \gamma^t \sum_{s \in S} R(s, \pi(b_t)) b_t(s) | b_0 = b\right] \quad (1)$$

A policy π which maximizes V^{π} is called an optimal policy π^*; it specifies for each b the optimal action to execute at the current step, assuming the agent will also act optimally at future time steps. γ here is the discount factor that trades off current rewards to future expected rewards ($0 \le \gamma \le 1$). The discount factor essentially determines how much the reinforcement learning agents care about rewards in the distant future relative to those in the immediate future.

3. STRAUSS

Fig. 2 presents the STRAUSS framework for scalable intent-driven industrial network service quality assurance with asset administration shells. We provide further details of the flow:

1. The production AAS is a digital twin of the production line (e.g. assembling mobile phones) in the form of AAS. Requirements may be input to the production AAS to modify the production rate, that must be autonomously handled by the robotic production and 6G network.
2. The first AI model, that utilizes POMDPs, maps the production rate to cycle times and energy efficiency targets for two categories of robots: Picking robots and Autonomous Mobile Robots (AMRs).
3. One important aspect of scalability is the ability to instantiate 10s to 100s of robots and manage them optimally within the framework. Through the efficient implementation of the robot asset administration shells, multiple robots can be managed without increasing computational load. In addition, an intent grouping agent is implemented that clusters the intents by equipment type and registered network services. This ensures that the system is not overloaded with multiple intents. The services considered are Simultaneous Localization and Mapping (SLAM), target detection and computational offloading.

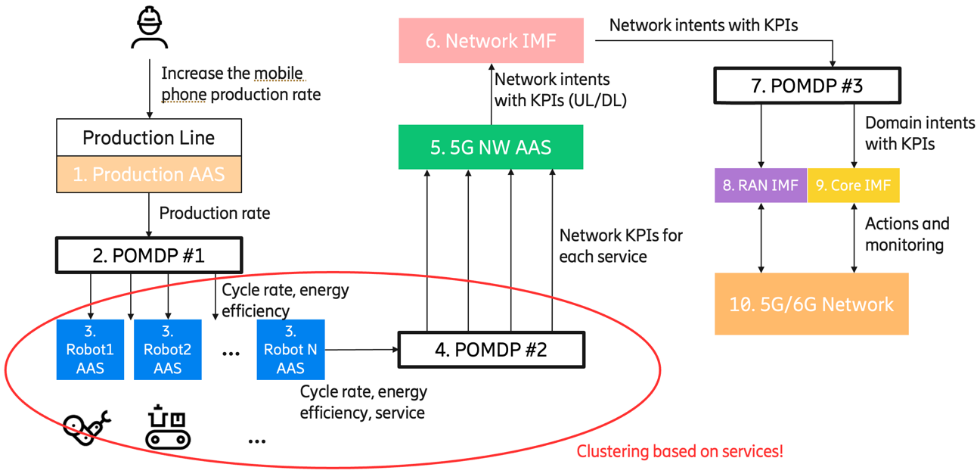

Figure 2 – STRAUSS framework

4. The second POMDP model maps the requirements of services to network key performance indicators. These could be in terms of throughput and latency for each service. An example of the typical ranges are provided in Table 2.
5. The 6G network AAS receives the service requirements and translates them into Resource Description Format (RDF) intent expectations to be handled by the network IMF [11].
6. The network IMF uses machine reasoning techniques [10] to determine intent issues. These issues are then decomposed as goals to be satisfied by the network IMF. To decompose expectations on throughput, energy and latency towards the hierarchy of IMFs, the network IMF invokes POMDP3.
7. This POMDP model uses QoS algebra to decompose throughput, latency and energy requirements. While latency and energy metrics are additive, throughput is the minimum value that must be met by the RAN and core IMF.
8. Decomposed expectations are provided to the RAN IMF that performs actions over the network. Actions such as UE handover or throughput rate limitation may be performed to meet targets.
9. Similarly, at the core IMF, actions such as setting 5QI values or scaling up/down the containers can be taken.

We will now look at further details on intent grouping agents, AI mapping and hierarchical intent decomposition.

3.1 Intent Grouping Agent (IGA)

Intent Grouping Agent (IGA) is one of the key components that enables large-scale operation with the STRAUSS framework. In a smart manufacturing environment, it is assumed that there exists multiple number of device types, that may require different industrial requirements. For example, an Automated Guided Vehicle (AGV) may require higher energy efficiency than a picking robot since AGV is operated on battery, whereas a picking robot is running on a power grid. We can also assume that devices in the same device type have the same industrial requirements; therefore their requests can be grouped by device type.

On the other hand, within the same device group, devices may subscribe to different network services, which may require different network KPIs. For instance, an AGV can be subscribed to a computational offloading service, where having a high data rate is critical; on the other hand, another AGV may be subscribed to a SLAM service, where low latency is critical. But the same device type with the same service is assumed to have the same network KPIs. Therefore, grouping can include service types as well.

In the STRAUSS framework, with the inclusion of IGA, before submitting to the IMF, we group network intents by device type and subscribed network service type. Supposed that there are N device types and M service types, then the number of distinct network KPI requirements becomes $N \times M$, instead of having network KPI requirements that are as many as the number of devices, which may be very large. The IGA is located between the device AAS's and POMDP2. Once grouping of intents is completed, IGA generates a requirement mapping in the form of JSON and sends the request to the POMDP2 for further processing. A sample IGA mapping is shown in Table 3.

3.2 QoS mapping

The example below provides an example of the POMDP format specification [12]:

Table 2 – 6G Service requirements

Service	Traffic	Throughput	Latency
Lidar	Periodic UDP	Uplink 64 kbps	10 – 100 ms
Camera stream	Periodic UDP frames	Uplink 0.5 – 20 Mbps	10 – 100 ms
Control traffic	TCP periodic	Downlink 7.5 kbps	10 – 100 ms

Table 3 – Intent grouping sample mapping

Robot	Simultaneous Localization and Mapping (SLAM)	Computational Offloading	Target Detection
AGV	Cycle: 32 Energy: 65	Cycle: 32 Energy: 70	Cycle: 20 Energy: 50

```
States (Unobservable):
#0 6G_SLA_satisfied
#1 6G_SLA_unsatisfied

Actions (6G Network Configuration):
#0 Latency _2_ms
#1 Latency _10_ms
#2 ThroughputDL_2_Mbps
#3 ThroughputDL _10_Mbps

Observations (Industry 4.0 Metrics):
#0 SLAM_Cycle_Time_10
#1 SLAM_Cycle_Time_30
#2 SLAM_Energy_Efficiency_50
#3 SLAM_Energy_Efficiency_70

Rewards (to be maximized):
R: SLAM_Cycle_Time_30  +20
R: 6G_SLA_satisfied    +30
```

The elements of this model include:

- *States*: The hidden state must be estimated by the POMDP model to ensure that the manufacturing process' Service Level Agreements (SLAs) are met (based on observed KPIs) while ensuring the 6G network optimizes network usage.
- *Actions*: The action space on the 6G network includes modifications in setting latency and throughput targets on 6G flows.
- *Observations*: Observations are provided by the manufacturing process, such as changes in SLAM service cycle time and energy efficiency.
- *Rewards*: The rewards are derived from the requirements and are to be maximized.

The POMDP problem file is input to solvers such as SARSOP [12] to provide a reward maximizing policy. Fig. 3 presents an example of a policy output with actions and observations responsible to move to an SLA-compliant state.

Table 4 provides a list of the POMDP models used within STRAUSS. POMDP1 is used to convert the requirements

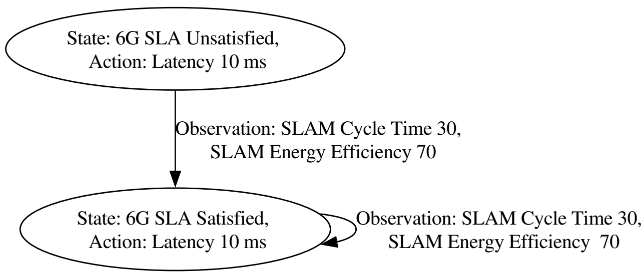

Figure 3 – POMDP policy example

Table 4 – POMDP models used in STRAUSS

Model	Input	Output
POMDP1	Production Rate	Robot Cycle time, Robot Energy Efficiency
POMDP2	Robot Services (SLAM, Computational Offloading, Target Detection)	Network Service Requirements on Throughput (UL/DL) and Latency
POMDP3	Network Service Requirements on Throughput (UL/DL) and Latency	RAN Throughput (UL/DL) and Latency; Core Throughput (UL/DL) and Latency

that are coming from the production AAS to picker and AMR robots. POMDP2 maps the services of the robots to network requirements such as throughput and latency. POMDP3 is invoked to decompose service expectations to RAN and core expectations. POMDP is a model-based reinforcement learning approach. We make use of the POMDP-solve format[2] that required an input of states, actions, observations and rewards.

The additional inputs that we use for POMDPs are the transition probability and observation probabilities.

```
T: <action> : <start-state> : <end-state> \%f
O : <action> : <end-state> : <observation> \%f
```

These probabilities are dependent on the use case and the datasets that are collected. For instance, in one of the POMDP models that we use, the action `cycle time 120s energy efficiency 70` can reach different transition and observation probabilities.

[2] https://www.pomdp.org/code/pomdp-file-spec.html

```
T: cycle_time_120s_energy_efficiency_70:
   network_optimal : network_suboptimal  0.3
O: cycle_time_60s_energy_efficiency_50:
   network_optimal: cycle_time_30s: 0.5
```

The models may be easily modified with new actions, observations and probabilities.

3.3 Hierarchical intent decomposition

The overall scalability of intent-driven service quality assurance highly depends on the intent hierarchy. Towards the vision of fully autonomous networks with the next evolution in the cellular domain, it is envisioned that the intents will be handled in a hierarchical manner.

In a Communications Service Provider (CSP) domain, the application of intent management function hierarchy realizes the decomposition of intents at each layer, towards the resources. Upon receiving a business intent, the business layer IMF creates services' intents, which are then translated into resource-level (e.g., RAN, core) intents for enhanced resolution and efficiency. This also helps to differentiate the scope of intents at each layer. It should be noted that the layers can be more than business, service and resource levels depending on the need. However, the implementation of this envisioned structure is not straightforward in a private network deployment. Especially when we consider the idea of end-to-end industrial automation with the integration of intelligent network management operations, the hierarchy of intents needs to be adjusted for smooth adaptation. Extending the scope of intents towards the industrial domain to handle enterprise expectations might require defining the scope of IMFs at each layer from scratch.

As depicted in Fig. 4, an industrial requirement, such as production rate, can be submitted by an enterprise operator or a machine to the business IMF. This is then decomposed into different service requirements to support the demand for an increased production rate. The holistic automation covering both the industrial domain and 6G system require the operations to be based on intent principles. Therefore, we expect to have both horizontal and vertical integration of IMFs, where vertical integration represents different layers and horizontal integration requires the interoperability of industry IMF and network IMF instances.

Even though the overall vision is to realize the digitalization of enterprises with end-to-end industrial automation, in this study we have a particular focus on IMF hierarchy in the private 6G network system. Towards this direction, we implemented the following IMF instances: (i) network IMF (ii) RAN IMF (iii) core IMF.

3.3.1 Network IMF

The exact interface of the network with the industrial domain is in between network 6G NW AAS and IMF. The network requirements gathered by 6G NW AAS is submitted to the network IMF in the form of intents. In this regard, an example of intent might include an expectation towards maximum latency or minimum data rate.

```
cc:robot-DL-throughput-intent
  a cc:Intent ;
  cc:hasExpectation
  [ a cc:MinMetricExpectation ;
      cc:target tel: robot _Premium_Service ;
      cc:params [ tel:averageThroughput 500];
  ] .
cc: robot -DL-latency-intent
  a cc:Intent ;
  cc:hasExpectation
  [ a cc:MaxMetricExpectation ;
      cc:target tel: robot _Premium_Service ;
      cc:params [ tel:averageLatency 100 ];
  ] .
```

Presented above is an example of the intent in Resource Description Format (RDF), which is compliant with 3GPP, TMForum. It has the following specifications:

1. Intent name: an identifier for human readability.
2. Expectations: expectations of observed metrics are expressed with semantics such as `MinMetricExpectation`: minimum value to satisfy the intent; `MaxMetricExpectation`: maximum value to satisfy the intent.
3. Target: target object for this expectation (e.g. network service or slice).
4. Params: metric and value for expectation (e.g. throughput).

Depending on the industrial use case, the scope of network Key Performance Indicators (KPIs) can be expanded accordingly. These network intents define the expectations, goals and constraints at the service level. These end-to-end service requirements should be decomposed into multiple network domains for enhanced scalability. In cases where the IMF is deployed as a centralized mechanism, the IMF should handle N different intents with end-to-end requirements. Furthermore, the conflict among these N intents becomes more challenging to be handled. When it comes to overall resource utilization, it is not a trivial approach to balance the load at the service level. Therefore, we need to create atomic expectations for simplified conflict resolution, where multiple instances are running in parallel. T7yu68"gfdhus, we can scale up resources when necessary. With atomic expectations at the lower levels, it becomes easier to detect and resolve the conflicts in an IMF.

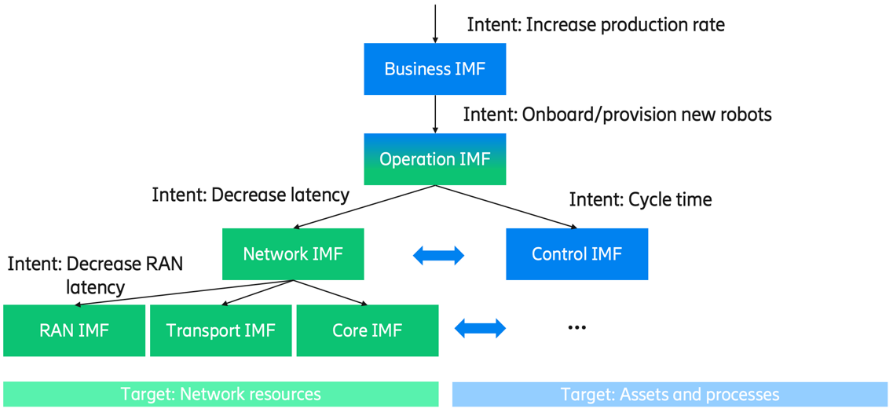

Figure 4 – Hierarchical intents for industrial 6G

3.4 Hierarchical IMF management

Within the network there are different resources and controllable actions. Towards the RAN, the spectrum can be divided within Physical Resource Blocks (PRBs). The PRBs may be allocated to services via partitioning or prioritized mechanisms to affect throughput and latency. Similarly, at the core, the resources can be the container capacities such as CPU, memory, disk. The container capacities can be scaled up or down to meet throughput and latency requirements.

As also specified by Ericsson [10], in the 6G vision we will have multiple IMFs coordinating to manage complex networks. As seen in the Fig. 5, the requirements from the network IMF are decomposed towards RAN and core IMFs.

The number of intents handled at each IMF can be decreased with the introduction of hierarchy. Additionally, scalability can be enhanced with the capabilities of load balancing and scaling up resources as well. In this direction, the network IMF can decompose intents into RAN and core IMF instances [9]. As mentioned, one example intent for AMR robots using an SLAM service can include the end-to-end latency requirement. This end-to-end latency requirement can be decomposed into RAN and core latency requirements separately for enhanced resolution. On one hand, it becomes easier to meet these intents in each domain. On the other hand, detecting and resolving intents at the resource layer is simpler considering the reduced scope at lower levels.

In our case, we consider conflicting scenarios as those generated by resource constraints. For say three intents, if the RAN is the bottleneck, by having this decomposition system, the RAN IMF can suitably resolve or escalate conflicts. However, if it were a monolithic IMF without root cause analysis, conflicts will have to be detected at both the RAN and core. Thus, the search complexity halves in this case. With increased levels of hierarchy, the

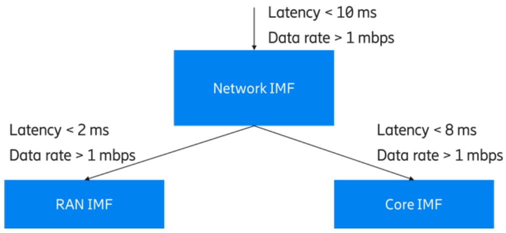

Figure 5 – Hierarchical intent decomposition

detection and resolution of conflicts will thus reduce. The distributed compute and reasoning abilities of individual IMFs also improves with this hierarchy.

By using a POMDP model, which has been introduced in a previous section, the network IMF can decompose latency requirements into RAN and core latency requirements. Since the end-to-end data rate requirement applies exactly in the same way to the resource levels, we do not require an intelligent mechanism for breaking down the data rate requirements. This POMDP model is responsible for intent decomposition, and the identification of RAN and core intents. An example is illustrated in Fig. 5.

Another important responsibility of the network IMF is to collect reports from RAN and core IMFs to evaluate if service intent has been met or not. So, it can be summarized that the action to be taken by the network IMF is intent decomposition, while monitoring scope consists of fetching reports from resource level IMFs.

3.4.1 RAN IMF

The IMF that is responsible for RAN measurements and configurations is called RAN IMF. Based on the intents sent by the network IMF, it starts handling its lifecycle.

Since the RAN IMF is one of the instances at the resource layer, it starts processing the intent upon receiving and configures RAN in order to meet the corresponding expectations. As exemplified, the RAN intent could consist of expectations towards maximum latency and minimum data rate to be achieved in this particular domain. It continuously fetches the throughput and latency measurements, and compares these with the threshold values specified in an intent. If there is an unmet expectation, the RAN IMF starts a closed loop operation until the expectation becomes satisfied [13].

In particular, in this paper we implement the RAN IMF that is able to take the following list of actions:

1. User Equipment (UE) handover to another gNodeB.
2. Modify Maximum Bit Rate (MBR) that a UE can achieve in RAN.
3. Configuration of the transmission power.

After the action proposal phase, the IMF starts predicting the impact of each action on the system state. Based on these predictions, the RAN IMF picks the action that maximizes the utility in the RAN IMF system. It should be noted that this paper implements a penalty-based mechanism to represent the utility of each action. In other words, the RAN IMF aims to minimize the total penalty, while actuating on the network.

In addition to latency and throughput expectations, the RAN IMF is implemented in a way that it supports another expectation towards energy consumption. The intent for enhancing energy efficiency in the network only targets the RAN domain, so the network IMF should directly relay such intent to the RAN IMF with the same content. So, the minimization of the penalty is not straightforward when there are multiple intents that might conflict. While the RAN IMF tries to increase throughput for meeting an intent, it may also increase the energy consumption of the network which is then creating a conflict. The conflict detection and resolution in the RAN domain is handled through the penalty-based evaluation mechanism.

3.4.2 Core IMF

The core IMF is the other resource level instance, which is responsible for handling the intents targeting the core domain in the network. While the first decomposed intent goes to the RAN IMF, the other decomposed intent is submitted to the core IMF for achieving a higher resolution in the core domain as well.

The capabilities of the core IMF is the same as for the RAN IMF at a high-level. While it monitors the measurements provided by the core domain of the network, it also accommodates relevant capabilities for reconfiguration when an intent is not met. Furthermore, for the network IMF to evaluate if end-to-end expectations of an intent is satisfied, the core IMF provides a reporting service to which the network IMF can subscribe for fetching the core network measurements and understand whether the decomposed intent is met or not.

In addition to the latency and throughput measurements in the core domain, the core IMF can reconfigure relevant functions. One of the actions to be taken by the core IMF is to modify the priority of a flow in the network, which implicitly defines the resources allocated to the corresponding service traffic. Moreover, increasing the priority of traffic impacts the scheduling decisions, which may decrease the queueing delay in the core domain.

In addition to the core network, we implemented another functionality to migrate the service context from a data center to another. For example, if a UE consumes a service provided at the edge, it is possible to move the UE context to the cloud. After this operation, the UE starts consuming a service from an instance deployed over the cloud. This operation might be valuable in decreasing the latency by avoiding the WAN access delay. Like the RAN IMF, the core IMF also implements a logical closed loop within. If a core intent is not satisfied (e.g., at least one of the expectations are not met), the core IMF starts the loop by proposing potential actions that can fix the issue. This set of actions can consist of moving UE to another data center and modification of the corresponding traffic flow priority. After the proposed new configurations, the core IMF starts predicting the impact of these actions on the system state to evaluate whether they can decrease the overall penalty or not. Based on these predictions (e.g., system state, KPIs), the core IMF determines the configuration that can minimize the penalty in the system and actuates on the network.

These closed loop operations [13] are executed until all intents are met or the system reaches steady-state. In a practical implementation, the core IMF and RAN IMF are expected to also implement escalation capabilities, so that the conflicts can be resolved at a higher level, with a larger scope. However, in this paper we only proposed decomposition and report interfaces between service and resource layer IMFs.

4. THEORETICAL ANALYSIS FOR SCALABILITY

The primary objective of the STRAUSS framework is to support intent handling capabilities of 6G and beyond networks' Intent Management Functions (IMFs). Although IMF deployments are designed to be both horizontally and vertically scalable with the help of elastic cloud deployments, improving network intent sources' efficiency may have many positive effects, such as efficient use of resources, energy efficiency, cost savings and many others. In this study, the proposed framework decreases the number of intents produced by individual devices within a smart factory environment by grouping industrial requirements by device type and subscribed service type. This results in the removal of redundant intents to be submitted to the IMF. Below, a theoretical analysis on message complexity of the proposed framework is given. We also provide a theoretical analysis of a typical framework without IGA for comparison.

As seen in Fig. 6, the process begins with a request towards production AAS. Then the production AAS relays this request to POMDP1. POMDP1 generates industrial requirements for each device type using MQTT pub-sub communication. This results in a scatter operation where each device receives only one message which is paired with its device type. This step involves in N message transfer where N is the number of devices. Then each device relays this message to the IGA, which again results in N message transfers. After the IGA gathers all messages from all devices, it generates one single message and sends this message to POMDP2. After POMDP2 processes this message, it sends the modified single message to NW AAS. NW AAS then generates NW intent messages for each group consisting of a specific device type and service type. This involves in $D \times S$ number of message transfer where D is the number of device type, and S is the number of service type. Therefore, the total number of messages generated in the STRAUSS framework is $\Theta(2 + 2N + (D * S))$. Moreover, the total number of intents that should be handled by the IMF is $\Theta(D * S)$.

In a typical 6G-enabled smart manufacturing environment, it is assumed that $(D \times S) << N$. Therefore, the STRAUSS framework decreases the number of intents to be handled by the IMF in a typical intent-based NW automation drastically.

5. EVALUATION

To justify the theoretical analysis and demonstrate the flow for the STRAUSS framework, we have implemented all the individual components as microservices. We have run end-to-end tests to observe the number of intent requests that are received by the IMF to justify the effects of having the IGA. In our test scenarios, we assume that we have two types of devices, AMR and picking robots. On the other hand, we assumed that the network provides three QoS services, SLAM, target detection and compute offload. We have run two test scenarios where the number of AMRs and picking robots are 10 and 30 respectively, which makes test scenarios with the total number of devices as 20 and 60. The robots are subscribed to the QoS services randomly. We have measured the total number of intents received by the IMF. We also run tests with similar scenarios without having the IGA for comparison. As shown in Fig. 7, when the IGA is used, the number of intents that are received by the IMF remains constant, as expected. On the other hand, when the IGA is not used, the number of intents increases as the number of devices increases.

Similarly, when multiple intents are received by an IMF, the IMF has to check for conflicting requirements or actions between the intent expectations. Taking pairwise comparison of combinations of intent expectations, this process may be characterized by the formula:

$$^{n}C_r = \frac{n!}{r! \cdot (n-r)!} \tag{2}$$

Where n represents the total number of intent expectations and r is set to 2. As seen in Fig. 8, without the IGA, the number of combinations to be checked increases significantly. This process of conflict evaluation would entail additional compute power and processing time at each IMF. Through the use of the IGA, these conflicts are reduced to only 15 cases, that may be evaluated in a deterministic manner. Additionally, through the use of network, RAN and core IMF instances, the intents may be hierarchically decomposed to ensure scalable and efficient intent management.

We reiterate that our technique is agnostic to the number of robots: 100, 500 or 1000. The Intent Grouping Agent (IGA) is responsible for grouping messages according to subscribed services and device types. While we have shown the outputs for two types of robots (AMR, picking) and three services (SLAM, compute offload, target detection), the system can be extended at scale to other combinations as well. Fig. 9 showcases the effect of both increasing the types of robots, as well as the subscribed services for each robot. For instance with three types of robots, with increased services of seven, only 21 messages are to be sent even with 100 robots. In a more complex scenario with nine types of robots and seven services each, the number of messages increases to 63. These messages can be handled in a scalable fashion, both by the AAS, as well as the IMF. Thus, we showcase that the system can scale for 100s of robots, types and subscribed services.

6. RELATED WORK

We provide a state-of-the-art analysis on industrial digital twins, 6G network service quality assurance and scalable network management.

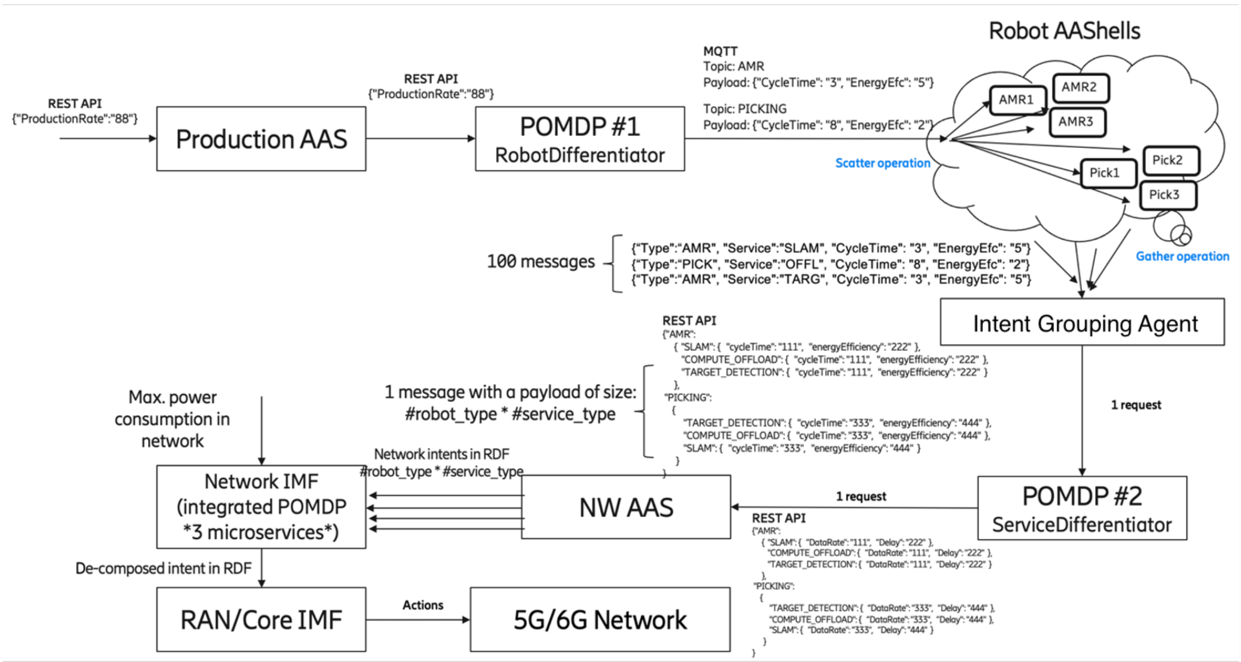

Figure 6 – STRAUSS framework sample flow diagram

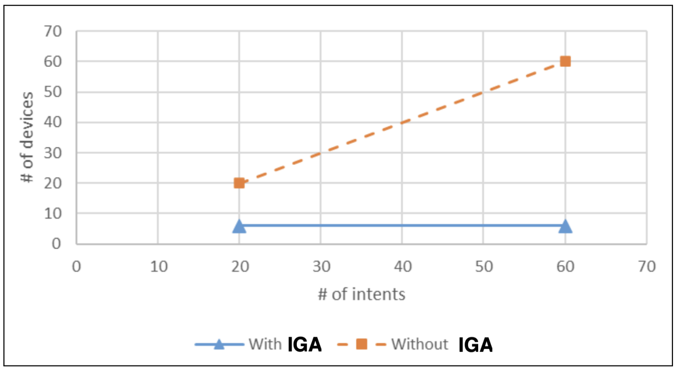

Figure 7 – Intents received by IMF

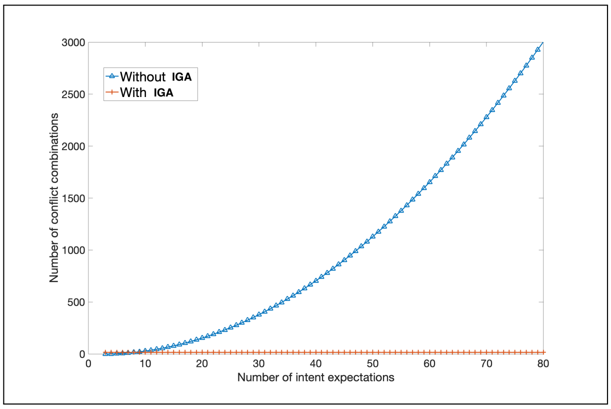

Figure 8 – Intent conflicts as a function of number of expectations

6.1 AAS

The Asset Administration Shell (AAS) is an industrial digital twinning technology that enables creation and management of the digital representation of Industry 4.0 assets [14]. The AAS is widely adopted by the Industry 4.0 ecosystem as a result of the wide standardization efforts carried by the industry ecosystem, such as Platform Industry 4.0, VDI and IEC [15, 16, 17, 18]. The high-level architecture of the AAS, is standardized by IEC [18]. VDI highlights the AAS as a key enabler for interoperability between I4.0 systems supporting various communication protocols. Besides the industry ecosystem, 5G Alliance for Connected Industries and Automation (5GACIA) works on efficient integration of 3GPP 6G systems in accordance with the underlying ideas of Industry 4.0, exploiting AAS [19].

6.2 Private 6G

The evolution of 6G networks has brought in new use cases across multiple domains such as XR, manufacturing, digital twins and healthcare [1]. To meet the challenges of indoor 6G, newer spectrum capabilities, edge computing and QoS management will be needed [20]. As specified in [6], these private networks can be implemented as completely self-contained standalone Non-Public Networks (NPNs) that have no connection to the public network, to NPNs that are hosted entirely by public network operators. AI native 6G [2] adds another dimension to autonomous management of networks with respect to industrial requirements.

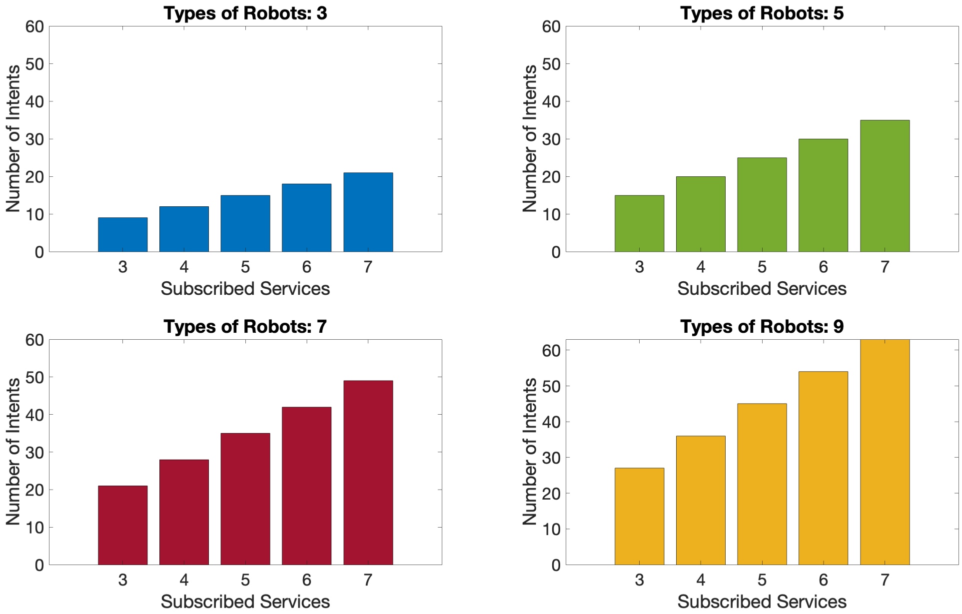

Figure 9 – Number of intents generated by the IGA with changes in robot types and subscribed services.

6.3 AI in network SQA

Industrial applications rely on the Quality of Service (QoS) of the underlying communications system, which has to meet the application requirements in each case. Some industrial use cases pose highly demanding communication requirements and are therefore quite sensitive to any changes in the QoS (e.g manufacturing, robots, automotive) [21]. 6G supports mechanisms for specifying and meeting QoS. One such technique is to use network slicing [22], that intelligently allocates network resources to meet differentiated QoS requirements. In [7], the authors provide a technique to map industrial requirements to slicing constraints for efficient 6G service quality assurance.

6.4 Intents for industrial networks

Intents are the formal representation of requirements to a system [3]. As specified in [23], an intent is associated with two parties: intent owner that creates and manages the intent; and intent handler that fulfils the requirements within an autonomous domain. Intents may be formally defined within RDF graphs as a way to structure intent models and extract their semantic correlation. In [24], a cognitive core solution is specified that can handle intent-driven specifications. In [25], a comprehensive analysis of the state- of-the-art in intent-driven networks including the intent description models, intent lifecycle management and a generalized architectural framework is presented. In [26], an overview of decomposing standardized intents towards autonomous domains is provided. The work in [8] combines intent-driven networking with an AAS to create an efficient technique to reconfigure industrial 6G networks. However, the limitation of scalability has not been considered in that approach. The use of intents in the industrial domains is highlighted in [27], with an extension of standard models proposed towards industrial domains. The use of intents within industrial supply chains with related architecture is presented in [28].

6.5 Scalable network management

Automated management of network resources requires efficient handling of requests and is expected to be held without human intervention in 6G networks. For this reason, the topic of autonomous handling Quality of Service (QoS) requests coming from the UEs is one of the hot research topics in the literature in this domain.

In [29], the authors highlight the difficulties in computing intent performance at scale during runtime. Machine learning approaches are proposed to predict performance changes. In [30], a scalable approach for intent-driven network monitoring and configuration is proposed. To handle requirements of compute workloads at scale, [31] proposes a 3-tier architectural framework with the serverless compute paradigm.

When multiple intent expectations are input into networking systems, the possibility of conflicting requirements increase. In [32], proposed is a conflict resolution mechanism in intent-driven open RAN deployments, by using deep neural network based on long short-term memory. In [33], a Quadruple-based Intent Conflict Resolution (QICR) engine is proposed to convert potentially conflicting intents to conflict-free intents. A negotiation framework for intent conflict resolution is provided in [34], whereby conflicting actors can negotiate for intent conflict resolution.

Andrea et al. [35] proposed a methodology for deploying an intent-based system for the computing continuum using server-less cloud services. This kind of system can be extended to manage network resources for 6G automation. Hatim et. al. proposed a framework featuring a distributed and AI-driven management and orchestration system for massive deployment of network slices in 6G in [36]. In [37], Zhang et al. proposed an intent-driven network and on-demand slice management framework for 6G. Wang et. al. introduced an Intent-driven Intelligent Task Scheduling Approach (IITSA), which models a Partially Observable Markov Decision Process (POMDP) and introduces a Multi-Agent Proximal Policy Optimization (MAPPO) method in [38]. Ahmet et. al. proposed a performance evaluation study for intent management in 6G networks in [14].

Although the topic of intent-based network automation for 6G has been studied heavily in the last decade, the scalability of QoS request handling is barely examined in the literature. To the best of our knowledge, we cannot find a study that focuses specifically on scalable intent handling when multiple intent requests are received by the network.

6.6 STRAUSS

Compared to prior state-of-the-art position, STRAUSS focuses on scalability aspects of industrial network service quality assurance. Via the grouping of intent requests using intent grouping agents, the number of intents are reduced significantly. In addition, implementation of the AAS is made more scalable through a separation of active and passive parts. The mapping of requirements between domains is ensured via a micro-service-based implementation of AI models. These features ensure scalability of the approach, with particular focus on industrial networks.

7. CONCLUSIONS

To handle the complexity of 6G industrial networks, a scalable and robust network service quality assurance framework is needed. In this paper, we propose STRAUSS, a framework to advance scalability of industrial networks by proposing features such as an intent grouping agent, hierarchical intent management, AAS deployment options and distributed AI-based QoS mapping. The system is deployed on a realistic use case involving tens of robots and is demonstrated to reduce the complexity in intent evaluation. STRAUSS will prove useful in future 6G industrial deployments.

Future directions include incorporating utilities for conflict handling within industrial network intents.

ACKNOWLEDGEMENT

This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) through the 1515 Frontier Research and Development Laboratories Support Program under Project 5169902 and has been partly funded in the framework of the Horizon Europe Project Target-X (Grant Agreement No:101096614), a project funded by the Smart Networks and Services Joint Undertaking (SNS JU) under Horizon Europe.

REFERENCES

[1] Ericsson. *Co-creating a cyber physical world*. Whitepaper. Ericsson, 2024.

[2] Ericsson. *Defining AI native: A key enabler for advanced intelligent telecom networks*. Whitepaper. Ericsson, 2023.

[3] D. Cooperson. *Intent in Autonomous Networks*. Research Report. TM-Forum, 2022.

[4] Platform Industrie 4.0. *What is an Asset Administation Shell from a technical perspective?* Report. 2021.

[5] J. Wee, M. Cho, Y. -S. Lee, and K. -T. Lee. “Design of Manufacturing Equipment Digital Twin Interworking Framework Based on REST API for Smart Factory”. In: *Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2024.

[6] 5GACIA. *5G Non-Public Networks for Industrial Scenarios*. Whitepaper. 5GACIA, 2021.

[7] A. Kattepur, A. R. Nair, M. Saimler, and Y. Donmez. “Industrial 5G Service Quality Assurance via Markov Decision Process Mapping”. In: *IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2022.

[8] A. Kattepur, A. C. Baktir, M. Saimler, D. Cokuslu, Y. Donmez, and A. Nair. “Coins: Cognitive Intent Based Service Quality Assurance via Industrial Asset Administration Shell”. In: *IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. 2023.

[9] L. Duboc, D. Rosenblum, and T. Wicks. “A framework for modelling and analysis of software systems scalability”. In: *Proceedings of the 28th international conference on Software engineering (ICSE)*. 2006.

[10] J. Niemöller, J. Silvande, P. Stjernholm, L. Angelin, and U. Eriksson. *Autonomous networks with multi-layer, intent-based operation*. Tech. rep. Ericsson Technology Review, 2023.

[11] 3GPP. *Intent driven management services for mobile networks*. Tech. rep. TR 28.312, 2024.

[12] H. Kurniawati, D. Hsu, and W.S. Lee. “SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces”. In: *Robotics: Science and Systems* (2008).

[13] Ajay Kattepur, Swarup Mohalik, Ian Burdick, Marin Orlic, and Leonid Mokrushin. "Convergence: Cognitive Intent Driven 5G Radio Access Network Slice Assurance". In: *IEEE Wireless Communications and Networking Conference, WCNC 2024, Dubai, United Arab Emirates, April 21-24, 2024*. IEEE, 2024, pp. 1–6. DOI: 10.1109/WCNC57260.2024.10570716. URL: https://doi.org/10.1109/WCNC57260.2024.10570716.

[14] Ahmet Cihat Baktir, András Zahemszky, Andrey Silva, Dagnachew Azene Temesgene, Dinand Roeland, and Mehmet Karaca. "Intent Manager Function with Actuation Predictions: A Performance Evaluation". In: *2023 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. 2023, pp. 276–281. DOI: 10.1109/MeditCom58224.2023.10266596.

[15] ISO/IEC/IEEE. *Systems and software engineering–vocabulary*. 24765:2017(E).

[16] Stefan Pollmeier Jörg Neidig Andreas Orzelski. *Asset Administration Shell Reading Guide*. Ed. by Plattform Industrie4.0. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/AAS-ReadingGuide202201.pdf?__blob=publicationFile&v=4.

[17] Karsten Schweichhart. *Reference Architectural Model Industrie 4.0 (RAMI 4.0)*. Ed. by Plattform Industrie4.0. URL: https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf.

[18] IEC 63278-1:2023. *Asset Administration Shell for industrial applications - Part 1: Asset Administration Shell structure*. Ed. by IEC. URL: https://webstore.iec.ch/en/publication/65628.

[19] 5GACIA. *Using Digital Twins to Integrate 5G into Production Networks*. Ed. by 5G Alliance for Connected Industries and Automation. URL: file:///C:/Users/ecokden/Downloads/5G-ACIA_Using_Digital_Twins_to_Integrate_5G_into_Production_Networks_single-pages%20(2).pdf.

[20] M. Wen and et al. "Private 5G Networks: Concepts, Architectures, and Research Landscape". In: *IEEE Journal of Selected Topics in Signal Processing* 16.2 (2022), pp. 7–25.

[21] 5GACIA. *5G QoS for Industrial Automation*. Whitepaper. 5GACIA, 2022.

[22] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck. "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions". In: *IEEE Communications Surveys and Tutorials* 20.3 (2018).

[23] TM Forum. *Intent Common Model*. Specification TR290 v3.6.0. 2024.

[24] J. Niemoller, L. Mokrushin, S. Mohalik, M. Konchylaki, and G. Sarmonikas. "Cognitive processes for adaptive intent-based networking". In: *Ericsson Technology Review* (2020).

[25] K. Mehmood, K. Kralevska, and D. Palma. "Intent-driven autonomous network and service management in future cellular networks: A structured literature review". In: *Computer Networks* 220 (2023).

[26] M. Xie, P. H. Gomes, J. Niemoller, and J. P. Waldemar. "Intent-Driven Management for Multi-Vertical End-to-End Network Slicing Services". In: *IEEE Globecom Workshops*. 2022.

[27] A. Baktir and E. Biyar. "Intent-Based Management for Industrial Automation". In: *Emerging Technologies and Factory Automation Conference*. 2024.

[28] M. Bensalem, J. Dizdarević, F. Carpio, and A. Jukan. "The Role of Intent-Based Networking in ICT Supply Chains". In: *IEEE 22nd International Conference on High Performance Switching and Routing (HPSR)*. 2021.

[29] Y. Wu, N. N. Mohanasamy, L. Jagadeesan, and M. R. Rahman. "Changes in Intent: Behavioral Predictions of Distributed SDN Controller Reconfiguration". In: *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 2021.

[30] Z. Xi and et al. "Newton: Intent-Driven Network Traffic Monitoring". In: *IEEE/ACM Transactions on Networking* 30.2 (2022).

[31] A. Morichetta, N. Spring, P. Raith, and S. Dustdar. "Intent-based Management for the Distributed Computing Continuum". In: *IEEE International Conference on Service-Oriented System Engineering (SOSE)* (2023).

[32] N. R. de Oliveira, I. M. Moraes, D. S. V. de Medeiros, M. A. Lopez, and D. M. F. Mattos. "An Agile Conflict-Solving Framework for Intent-Based Management of Service Level Agreement". In: *2nd International Conference on 6G Networking (6GNet)* (2023).

[33] J. Zhang and et al. "A Conflict Resolution Scheme in Intent-Driven Network". In: *IEEE/CIC International Conference on Communications in China (ICCC)* (2021).

[34] Y. Sharma, D. Bhamare, A. Kassler, and J. Taheri. "Intent Negotiation Framework for Intent-Driven Service Management". In: *IEEE Communications Magazine* 61.6 (2023).

[35] Andrea Morichetta, Nikolaus Spring, Philipp Raith, and Schahram Dustdar. "Intent-based Management for the Distributed Computing Continuum". In: *2023 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. 2023, pp. 239–249. DOI: 10.1109/SOSE58276.2023.00035.

[36] Hatim Chergui, Adlen Ksentini, Luis Blanco, and Christos Verikoukis. "Toward Zero-Touch Management and Orchestration of Massive Deployment of Network Slices in 6G". In: *IEEE Wireless Communications* 29.1 (2022), pp. 86–93. DOI: 10.1109/MWC.009.00366.

[37] Jingwen Zhang, Chungang Yang, Ru Dong, Yao Wang, Alagan Anpalagan, Qiang Ni, and Mohsen Guizani. "Intent-Driven Closed-Loop Control and Management Framework for 6G Open RAN". In: *IEEE Internet of Things Journal* 11.4 (2024), pp. 6314–6327. DOI: 10.1109/JIOT.2023.3312795.

[38] Qingqing Wang, Sai Zou, Yanglong Sun, Minghui Liwang, Xianbin Wang, and Wei Ni. "Toward Intelligent and Adaptive Task Scheduling for 6G: An Intent-Driven Framework". In: *IEEE Transactions on Cognitive Communications and Networking* 10.5 (2024), pp. 1975–1988. DOI: 10.1109/TCCN.2024.3391318.

AUTHORS

Deniz Cokuslu is a senior researcher in the Ericsson Research Networks group. His expertise includes, but is not limited to, asset administration shells, network automation, network exposures and distributed systems. Before Ericsson, Dr Cokuslu worked for academics in software and computer engineering departments. He received his BSc and MSc from Izmir Institute of Technology, Department of Computer Engineering, and his PhD from Paul Sabatier University, EDMITT.

Ajay Kattepur is a senior researcher in the Autonomous Intelligent Systems group at Ericsson Research, with a focus on applying automated planning, reinforcement learning and verification techniques for 5G networking and robotics applications. Prior to joining Ericsson Research, he was with Tata Consultancy Services (TCS) Research and Innovation Labs in India. Ajay received his Ph.D. in Computer science from the French National Institute for Computer Science and Control (INRIA), Rennes, France, and M.Eng. and B.Eng. degrees in electrical and electronic engineering from Nanyang Technological University (NTU) Singapore.

