Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 137 (2018) 90–101

Procedia Computer Science

www.elsevier.com/locate/procedia

SEMANTiCS 2018 – 14th International Conference on Semantic Systems

Ensuring Workplace Safety in Goal-based Industrial Manufacturing Systems

Ahmed Shafei[a,b], Jack Hodges[c], Simon Mayer[c,d]

[a]BSH Hausgeräte GmbH, Stuttgart, DE
[b]Technical University of Munich, Munich, DE
[c]Siemens Corporate Technology, Berkeley, USA
[d]Pro2Future GmbH and Graz University of Technology, Graz, AT

Abstract

One of the most critical challenges in human-robot collaborative work settings is ensuring the health and safety of the involved human workers. We propose to integrate task-level planning with semantically represented workplace safety rules that are published by regulatory bodies, meaning that our system can adapt to produce different variants of a product while respecting workplace safety regulation. Our prototype system interacts with human workers and machine agents via Activity Streams and a speech synthesis interface, and we have shown that its SPIN reasoning engine can scale to scenarios that incorporate complex products and many agents. The current system state and action logs of the agents and products are easily observable using a dashboard interface. The semantic models were evaluated by five experts in workplace safety and process engineering who expressed confidence about using, maintaining, and even extending the models themselves after only negligible training, a crucial factor for the real-world adoption of such systems.

Keywords: Workplace Safety; Automated Planning; Semantic Web; SPIN Reasoning; Web of Things; Internet of Things

1. Introduction

Driven by academic research, the Internet of Things has gone beyond basic enabling technologies for the interconnection of smart devices, and researchers are increasingly focusing on the higher-level interoperability of Internet- and Web-enabled devices and their services: one of the core challenges for ubiquitous computing research, today, is

* Ahmed Shafei. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: ahmed.shafei@bshg.com

10.1016/j.procs.2018.09.009

supporting users toward creating meaningful combinations of physical services for intelligent systems [1]. In industrial settings, one of the forms that this challenge takes is how to enable the rapid reconfiguration of manufacturing environments, one of the core enablers of a *lot-size-one* world where unique products are created at mass quantities and mass-production cost and prices [2]. This depends on production resources – including machine and human agents and possibly the products themselves – being aware of each other's capabilities, limitations, strengths, and weaknesses, and acting collaboratively within the context of that knowledge. At the core of this capability is a common conceptual model of agency, action, and process that can be applied equally well to human and machine workers, and which can be used to explain why an agent is (or is not) chosen for a work task when planning collaborative processes. However, the planning of human-robot collaborative (HRC) assembly processes needs to take into account not only functional properties (i.e., whether an agent can accomplish a specific industrial action, and how) but also *non-functional* ones: given an agent's specific skills, preferences, and ergonomic capabilities, *should* that agent take part in a specific assembly process, and in what capacity?

One of the most critical non-functional aspects of workflow planning, particularly with respect to human agents in HRC settings, is ensuring the safety of the workers involved: any job performed in the workspace may have potential hazards that can affect the human workers or the workspace as a whole (e.g., *grinding* incurs a *noise hazard* that can cause *hearing loss*). Therefore, *Job Hazard Analysis* (JHA) is used to identify potential hazards and ways to mitigate them (e.g., by *wearing ear protectors*) [3]. The implemented mitigation strategies have to be compliant with workplace safety regulations, which are established and enforced by regulatory bodies such as the United States *Occupational Safety and Health Administration* (OSHA). OSHA penalties can result in fines of up to $70,000 for each violation, blacklisting for the involved company, and prison time for responsible persons.

Traditionally, a process engineer is responsible for assigning tasks to the most suitable workers to ensure the functional and non-functional optimality of an industrial process. Workplace safety regulations are met through offline JHA of tasks performed, explaining identified potential hazards to human workers, and having them acknowledge the JHA process by signing a paper form that explains hazards and mitigation procedures.[1] However, businesses struggle with the implementation of safety and health rules because regulations are perceived as overly complex and enforcement and documentation require high investment, a challenge that is aggravated when production processes start involving dynamically planned workflows and that, in effect, hinders the real-world implementation of flexible workflow planning.

In this paper, we present a generic framework and prototype implementation that enable the *safety-aware planning of industrial workflows*, thereby opening the possibility to automate JHA by drawing on machine-readable versions of safety standards such as those created by OSHA: our systems *automatically enforce workplace safety law* according to knowledge about OSHA regulations that is stored inside a domain ontology and in addition document employee compliance with these regulations. We accomplish this by using semantic reasoning together with a number of ontologies that capture relevant conceptual knowledge from the domains of industrial production, workplace safety, and medicine. The workflow planning happens in near real-time and without manual intervention while the system's knowledge inside our domain ontologies is kept current by means of real-time sensor data acquisition. When planning, the system not only adheres to workplace safety norms, but also adapts its behavior to other preferences (e.g., their handedness), capabilities (e.g., their skills and certifications), and restrictions (e.g., ergonomic) of available agents. Finally, we have shown that the domain knowledge that our solution utilizes can be used, maintained, and even extended by subject matter experts (i.e., workplace safety experts and process engineers) without programming skills and with only negligible prior training.

We demonstrate the real-world applicability of our approach in the context of dynamically creating and executing collaborative, safety-aware production plans for customized furniture. We focus on the collaborative assembly of furniture because this domain seems particularly appealing for declarative manufacturing systems due to its highly customizable goods and high customer demand for customization. Our system was deployed in the context of a prototype manufacturing cell in our laboratory (see Fig. 1) that contains a *Universal Robotics UR5* robot that is controlled through the Robot Operating System (ROS), a *fischertechnik* toy robot that is interfaced through a programmable logic controller (PLC), several devices that represent production machines, and a *Microsoft Kinect* device and loudspeaker for interacting with human workers.

[1] An example JHA form can be accessed at `http://www.safetyworksmaine.gov/pdf/haz_samples.pdf`

Fig. 1: Our laboratory manufacturing cell with human and robot agents.

In the following, we discuss related work on ontology-based knowledge modeling of human and machine interaction/collaboration and point out where our approach advances the state of the art. In Section 3, we present our approach for enabling the safety-aware planning of collaborative industrial workflows; this includes a detailed discussion of the information models that form the basis of our system and of its user frontend. In Section 4, we show the results of an evaluation of our prototype system with subject-matter experts in industrial manufacturing to demonstrate its applicability in real-world scenarios, and discuss the scalability of our system.

2. Related Work

Considering humans and machines as collaborators in a factory context requires a model of the kinds of processes that they can perform and the constraints that limit their participation in processes. Early work on modeling human intentionality in actions, plans, and goals was done by Schank and Abelson [4] in the context of natural language understanding. In modern systems, procedures that can be executed by humans have been modeled using procedure and step models such as the *Process Specification Language (PSL)* [5], *ISA-88* [6], the *Procedural Representation Language* [7], and activity models (e.g., Activity Streams[2]) – the *KnowRob* system uses an action model [8] to describe procedural recipes.

Agent activities can be constrained in any number of ways, including physiological constraints, societal constraints, situational constraints, and preference constraints. Saad et al. [9] studied constraints on human movement in terms of ergonomic positions, Reed et al. [10] constructed ergonomic models of various human shapes using a Microsoft Kinect device, and Wiendahl et al. [11] describes factory planning requirements that help identify relevant human activities and norms – in this work, we propose to put these requirements in a machine-readable form and interconnect them with additional ontologies to enable dynamic planning in HRC settings; we also draw upon the exploration of kinematic models for human users in HRC scenarios from Bestick et al. [12]. In the U.S., the Department of Labor prescribes limitations/constraints on human activity in its standard on Workers Rights.[3] [3] reports on an ontology and set of rules to help identify OSHA violations in construction and to suggest remedies by ontology-based JHA. The resulting construction safety ontology enables more effective inquiry of safety knowledge, but the authors do not consider the automatic implementation of that knowledge – in this work, we suggest that this information should already be used in the *production planning* phase, thereby enabling the upfront avoidance of hazards.

To enable collaborative planning in HRC scenarios, agent activities need to be modeled in a manner that allows human workers and machines to be swapped into or out of a role without regard to anything but their capabilities and constraints. To enable this, activity models need to be described at a high abstraction level, but which also can be mapped to low-level machine motion planning, for example by using the CRAM planner [13]. The high-level recipe model described in [13, 8] however is a purely robotic agent model where the planning happens at the machine motion planning level. In contrast, we have developed a collaborative operational model that combines the notion of a recipe (as a high-level goal or task, a bill of materials, and a sequence of steps), and an agent model with capabilities and constraints that can be applied equally to both human and robotic agents. Combining these models with a hierarchical

[2] https://www.w3.org/TR/activitystreams-core/

[3] OSHA 3021-2016, https://www.osha.gov/Publications/osha3021.pdf

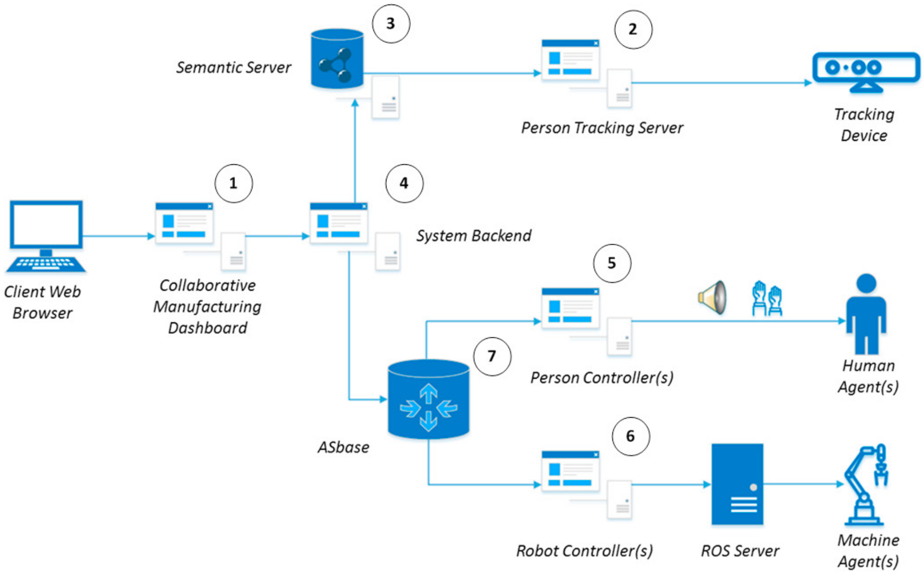


Fig. 2: Overview of the components of our prototype implementation.

planner, a means of acquiring real-time position data, the ability to communicate with the human worker, and an easily usable supervision interface comprises our system.

3. System Architecture

Our prototype system enables factory managers to configure a desired piece of furniture using a Web-based user interface and then instruct the planning component to produce a collaborative (human-robot) manufacturing plan for assembling the configured item. Importantly, the planning component incorporates different kinds of constraints that are expressed in our system's ontologies for HRC environments, including capabilities of agents and constraints regarding workplace safety – based on these constraints, it mitigates workplace safety hazards already during planing: for instance, it might insert the instruction to *wear anti-vibration gloves* for human workers before they are instructed to *operate high-powered drills*. Upon successful planning, the assembly plan is displayed and the operator can decide to execute it. Finally, the system's dashboard displays an overview of all successfully mitigated health and safety hazards, and this information is persisted in the system for documentation purposes.

Our system consists of seven basic components: The *Collaborative Manufacturing Dashboard* (at Fig. 2, marker **1**) is our system's user front end. It was designed for factory managers and is used for configuration, monitoring, and control tasks. The dashboard can, for instance, be used to add additional human and machine agents to the system, configure new pieces of furniture, and start production processes. We discuss this component in greater detail in Section 3.1. The *Person Tracking Server* (at **2**) is responsible for tracking human agents in their workspace. In our setup, real-time position information of an agent's body parts and joint angles are obtained from a *Microsoft Kinect* device and associated with human agents that are known to the system by means of SVM classifiers that are trained on the agents' physiologies. This information is shared with the *Semantic Server* (at **3**) that is responsible for storing and managing all ontologies (see Section 3.2) and includes a *SPARQL Inferencing Notation* (SPIN) engine for model-based reasoning.[4] The *Semantic Server* also receives input from the *Collaborative Manufacturing Dashboard* (information about added furniture pieces, registered agents, etc.) via the *System Backend* and continually integrates this new information with its representation of state.

[4] For an extensive in-depth discussion of this component, including its ontology management, access, and visualization features, we refer interested readers to [14]

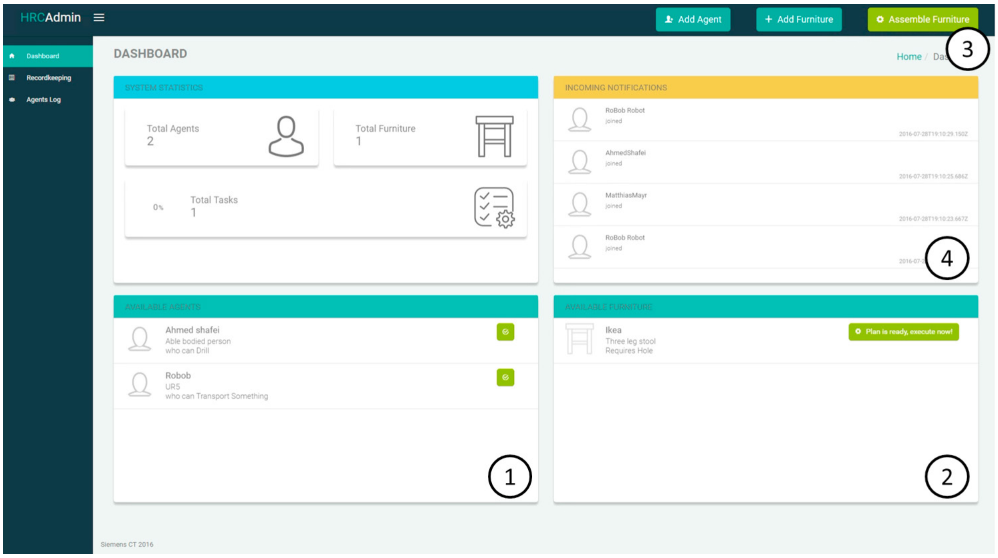


Fig. 3: Our system's Collaborative Manufacturing Dashboard is used to manage agents and furniture products (at **1** and **2**), trigger the planning and execution of manufacturing workflows (at **3**), and monitor the execution of assembly tasks in real time (at **4**).

When executing production plans, the *System Backend* (at **4**) communicates with several control modules for the different agents: For human agents, we use so-called *Person Controllers* (at **5**) that are created and destroyed on demand by the *Person Tracking Server* and communicate with workers using verbalized instructions via a speech synthesis interface. Workers can acknowledge commands using simple gestures (i.e., by raising their hand) and via a speech recognition component (i.e., by saying *Ok!*). For machines, for instance our UR5 robot, commands are executed via *Robot Controllers* (at **6**) that provide a REST API to provided functions (e.g., moving the robot arm and actuating the gripper). For communication between these components, we use a custom-built *Activity-Streams*-based messaging middleware called *ASbase* (at **7**; [15]). The *System Backend* links the *Collaborative Manufacturing Dashboard* with this middleware and the *Semantic Server*.

3.1. Collaborative Manufacturing Dashboard

Our system's *Collaborative Manufacturing Dashboard* (see Fig. 3) is a tool for factory managers that serves three main purposes:

Monitoring: The dashboard visualizes all relevant information from the *Semantic Server*. This includes data about known agents and furniture pieces as well as a log of *ASbase* messages which enables tracking the execution process of assembly tasks in real time. In addition, the dashboard shows statistics about the known agents (current status, number of actions carried out, total working time, etc.) and a log of their past actions. The dashboard also calculates a (mocked) salary/cost for each agent that is based on the value of all actions that agent has performed.

Configuration: Operators can use the dashboard to add new human and machine agents to the system, and for creating and modifying furniture pieces. For example, to add a new worker to the system, the operator enters this person's basic information including preferences such as the person's handedness, and the person's skill set. The *Person Tracking Server* then trains a new SVM classifier for identifying that agent using input from the Kinect device (alternatively, an existing classifier can be used). This information is then sent to the *Semantic Server* that updates the system's information model. Adding new furniture pieces to the system happens in a similar way.

Execution/Control: Finally, the dashboard enables an operator to select pieces of furniture to be assembled and to request an assembly plan from the *Semantic Server*. Upon successful planning, it displays the steps required for assembly together with the agents it will instruct to carry out the assembly sequence (see Fig. 4). After the assembly plan has been generated, the operator can decide to execute it. During execution, the dashboard updates the operator about the ongoing production process. The system can also recover from exceptions – for instance, if a required worker

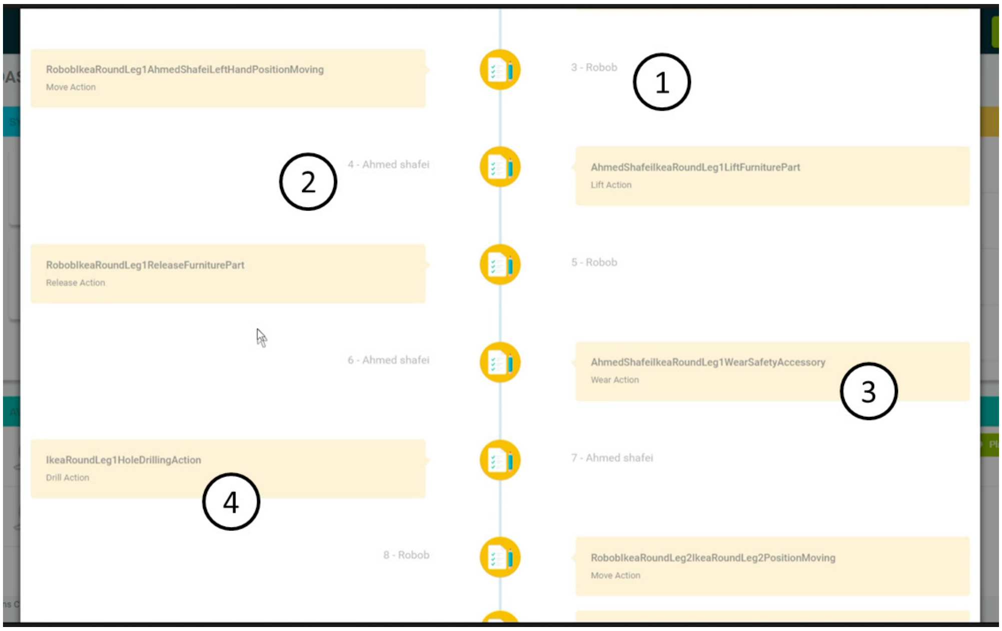


Fig. 4: A generated assembly plan for a three-legged stool. The system plans for collaborative workflows by human and machine agents (at **1** and **2**) and mitigates health hazards by inserting actions that ensure workplace safety. In this case, the drilling task (at **4**) incurs a *Vibration* hazard that is mitigated by instructing the worker to wear anti-vibration gloves (at **3**).

leaves his workplace, the system instructs his associated *Person Controller* to ask him to return into the tracker's field of view. After the assembly process has finished, the dashboard displays a success message that includes an overview of the health and safety hazards that were mitigated by the system. In this case, the worker was instructed to wear anti-vibration gloves during drilling which mitigated the *Vibration* hazard for his/her hands and arms. According to OSHA, this potentially helped avoid two health problems: *Hand-arm Vibration Syndrome (ICD-10 I73.9)* and *Carpal Tunnel Syndrome (ICD-10 G56)*. In addition, the system avoided *Back Stressing* by having the robot lift a heavy workpiece from the ground instead of asking the worker to perform this action.

3.2. Ontological Modeling of Manufacturing Processes and Safety Regulations

To enable safety-aware goal-driven workflow planning in HRC environments, our system depends on several ontologies that contain information about the ergonomic constraints of different agents, hazards, and mitigation strategies (at **1** in Fig. 5) as well as tools/equipment for workplace safety (at **2**) and positional information about human agent body parts and joints (this is necessary to relate data from the perception system to the agents in our system, at **3**). In addition, we created a capability model for collaborative workspaces that contains information about (human and robotic) agent skills (at **4**). Our models are interoperable with other widely used ontologies such as the Quantity model in QUDT (at **5**)[5], as well as with international standards that address batch process control such as ANSI/ISA-88, and widely accepted definitions of production processes such as DIN 2860 and DIN 8580 (at **6**). The standards our models are built on top of are shown alongside our models in the figure, in bold. The integration models and some glue models, such as FONM [16], which represents mechanical function and use, are also depicted in the figure.

Our system's *Capability Model* contains causal information about capabilities that can be carried out by agents. In the model, we differentiate between several types of agent capabilities: *Manipulations-of-Self* include ergonomic ones (e.g., bending a joint), translational ones (i.e., moving), and other capabilities that are relevant to workplace safety (e.g., wearing a hard hat). *Manipulations-of-Other* include industrial and translational modifications of workpieces.

[5] http://qudt.org/

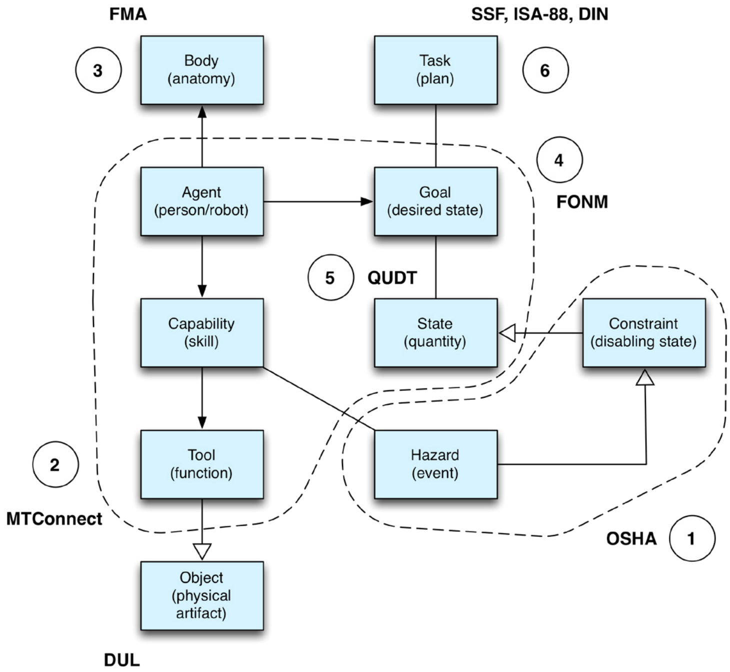


Fig. 5: High-level Information Model Architecture.

These correspond DIN 2860 definitions, such as *pressing on/in* and *bonding* for industrial modifications, and *passing on* and *placing/orienting* for translational ones. The central component of the *Capability Model* when executing a production sequence is the `Capability` class: this class represents the act of an agent performing an activity to achieve a goal. Instances of this class contain links to the specific URLs that are invoked to start executing an activity, and also hold references to ontology-embedded SPIN functions that are used to generate verbal instructions for communicating with human agents. Each capability is furthermore associated with the necessary body parts and tools for performing the activity as well as with hazards that are incurred by the activity. These associations point to the following four separate information models:

- The *Body Model* describes the anatomical structure of agent bodies including body parts (e.g., arms and legs) and their associated body joints (e.g., elbows and knees) and is aligned with the *Foundational Model of Anatomy*[6]. It also captures positional information (i.e., 3D positions and angles) about a specific agent's body parts and joints, which is provided to the ontology in real time using appropriate perception equipment: a body tracker camera is used for human agents, and the robotic agent in our setup exposes an API that enables querying its current state. The *Body Model* has five root classes: The `Part` and `Joint` classes represent body parts that are associated with an agent's body. The `Position` and `JointAngle` classes represent 3D positions and angles of body parts/joints. Finally, the `TrackerLabel` class defines labels for the joints and body parts.
- The *Agents Model* represents known agents, their capabilities (by linking to the *Capabilities Model*), and their ergonomic properties (by linking to the *Body Model*). This enables the system to reason about the capabilities and ergonomic constraints of different agents, for instance *one-armed robots*, *lower-limb disabled persons*, or *able-bodied persons*. Furthermore, this model contains preferences of the different agents – for instance, it records their preferred language for voice-based interaction with the agent.

[6] http://si.washington.edu/projects/fma

- For satisfying regulatory constraints with respect to workplace safety, the system needs to be able to reason about the tools that are used when an agent performs an action, and their properties – for instance, low-vibration drilling can be performed by human agents for longer periods than high-vibration drilling. In this case, the system often needs to identify corresponding safety accessories that can be used to avoid hazards associated with using a specific tool (e.g., anti-vibration gloves can be used while drilling). This information is captured in the *Tools Model*.
- The *Hazards Model* was derived from OSHA workplace safety regulations and formally defines regulatory constraints in terms of operational hazards, their associated risks, and possible mitigation strategies that are suggested by OSHA. Hazards and strategies are connected to potentially affected body parts in the *Body Model* and to agent capabilities in the *Capabilities Model*. The *Hazards Model* has three important root classes: The `Hazard` class represents safety hazards such as *Vibration* or *Back Bending*. Hazards are classified in seven hazard categories based on the OSHA classification: *biological*, *chemical*, *ergonomic*, *physical*, *psycho-social*, and *safety*. Hazards impose risks on agents (e.g., hearing loss) and their environment (e.g., fire) – these are captured in the `Risk` class. Finally, hazards are linked to mitigation strategies that are represented using the `Mitigation` class. Mitigation strategies are in turn associated with the maximum permitted exposure time to an underlying hazard, and (optionally) with safety accessories in the *Tools Model*.

Summarizing, our ontologies can represent agents in an industrial context and link their capabilities to body parts, tools, and workplace safety hazards, risks, and mitigation strategies. Our ontologies are flexible enough to implement a wide range of OSHA rules, including regulation regarding maximum usage times of certain equipment (per day/year/overall) and enforcing ergonomics-related rules, for instance regarding crouching and kneeling. All *Avoidance* and *Mitigation* actions that our system takes to ensure workplace safety are explicitly stated in a status message that it displays after an assembly plan has been executed.

For the real-world applicability of these models, it is important that subject-matter experts in production engineering and workplace safety can relate to the terminology and structure of the ontologies – to ensure this, we made sure to closely align those parts of the models that are most visible to the outside with widely used industrial standards: the *Capabilities Model* is aligned with DIN 2850 to ensure that it can be easily understood by production engineers and the *Hazards Model* re-uses OSHA terminology to facilitate its usage by workplace safety experts.

3.3. Safety-aware Flexible Workflow Planning

After initialization of the back-end and sensing components of our system and loading of our ontologies by the *Semantic Server*, the *Person Tracking Server* continually monitors our prototype manufacturing cell and uses the Microsoft Kinect's outputs together with the SVM classifiers that are trained during agent creation to identify currently available workers – this information is updated in the system's information model by the *Semantic Server*. As soon as a new manufacturing request is issued via the *Collaborative Manufacturing Dashboard*, a new task is inserted in the information model that describes the furniture to be assembled and the *Semantic Server* invokes its reasoning engine to plan the assembly task. The reasoning depends on SPIN rules that can be categorized into four categories:

Individual Constructors are SPIN rules that initialize variables and relationships when new class individuals are inserted in the Agent and Body ontologies. *Real-time Data Acquisition Rules* are SPIN rules that dynamically query real-time sensor data about agent positions and joint angles and aggregate it into the system's information model.[7] These rules are invoked on-demand when sensor input is required. *Activity Stream Creation Rules* are used to dynamically generate activities in the Activity Streams format for dissemination to the Robot/Person controllers.

Finally, *Assembly Task Planning Rules* are used for planning furniture assembly tasks while satisfying all planning constraints in a given scenario. This process consists of several SPIN rules that together initialize a new sequence of activities and populate this sequence with agent actions until the assembly task has been satisfied. For each action, the agent is selected while considering capability, physiological, and positional constraints on the agent (using the *Real-time Data Acquisition Rules*). Subsequently, the regulatory constraints are evaluated for the current agent action

[7] This is done by using SPIN JavaScript Functions that define JS-based SPARQL functions that are executed at runtime, see `http://spinrdf.org/spinx.html` for more information.

and the system mitigates workplace safety hazards by inserting safety-relevant actions directly into the workflow. This is one way of dealing with workplace safety hazards that it supports – re-using OSHA's vocabulary, we refer to this strategy as *Mitigation*: the action that incurs the safety hazard persists in the assembly plan (in this case, performing the drilling task), but the safety hazard has been mitigated by means of explicitly extending the plan with a new action that in this case motivates the worker to wear a safety accessory. The system also supports another approach, referred to as *Avoidance*: for instance, since OSHA discourages the lifting of items heavier than 50lbs/~23kg by human workers, our system avoids heavy lifting actions if a robot is available that can take over. *Avoidance* leads to an implicit adaptation of the assembly plan.

After the reasoning process has finished, the system uses the *Activity Stream Creation Rules* to generate the applicable tasks for workers and robotic agents – in the case of human workers, these Activities also include a verbalized instruction. These tasks are then issued via HTTP requests to the *ASbase* platform which distributes them to applicable subscribers (i.e., *Person Controllers* and *Robot Controllers*). The system back-end drives this process by relying on updates that are relayed through *ASbase*, most importantly acknowledgements from agent controllers. The system also reacts to exceptions during the implementation of a manufacturing plan: for instance, human workers who leave the manufacturing cell are dynamically tasked to return to the field of view of the Kinect device.

4. Evaluation

We evaluated our prototype system with respect to its practicability and scalability: by conducting interviews with five domain experts, we investigated the system's compliance with current workplace safety recommendations and its extensibility by subject-matter experts (see Section 4.1). Furthermore, we measured how the performance of the planning component in our prototype implementation scales with an increasing number of agents and increasing complexity of the assembled products (see Section 4.2).

4.1. Expert Evaluation: Practicability, Extensibility, and Ease of Use

To evaluate the real-world impact and practicability of our approach, we ideally would be able to provide quantifiable metrics to assess the system's performance directly, such as statistics on the weekly or monthly accident rates before and after the system implementation. Since, however, it is (almost) impossible to obtain this kind of data for a research prototype, we decided to conduct one-on-one interviews with domain experts from one of the world's largest industrial manufacturing companies and ask them what impact they would *expect* from our proposed system. In their day-to-day work, the three interviewed *Safety Experts* are responsible for the safety of workers and the workplace environment in production facilities and are, in general, familiar with workplace safety recommendations and practices. The two interviewed *Process Engineers*' work involves the development, configuration, and optimization of industrial processes and they are, in general, knowledgeable about the planning and optimization of workflow processes.

Each expert interview (about 30min) consisted of four steps: We started by briefly introducing semantic technologies in general as well as the implemented system and its main ontologies. This introduction was followed by in-depth discussions about the concrete implementation of our prototype system and its ontologies, using graphical depictions of the models as visual aids – in particular, we covered (a) the *Workplace Safety Model*, including examples of physical hazards and their mitigation techniques, examples of ergonomic hazards and their mitigation techniques, and examples of risks and (b) the *Manufacturing Assembly Model*, i.e. the result of the integration of all top-level concepts across the knowledge models used within our system. Finally, we showed a live demonstration of the system using the *Collaborative Manufacturing Dashboard* to generate a collaborative assembly plan for a 3-leg stool using two agents: a UR5 robot and a human worker. After each interview, we collected additional feedback from each interviewee using a survey with five single-choice questions (5-Point Likert scale) and a free-text feedback box.

Our study participants responded that they were familiar with the hazard and safety concepts that are contained in the *Workplace Safety Model*: four of them were *Very Familiar* or *Extremely Familiar* with the modeled concepts, and only one interviewee indicated being *Somewhat Familiar* with them. The interviewees thus had little trouble when *reading and traversing* the *Workplace Safety Model* (see Figure 6, question **a**), which indicates that the ontology's hierarchical taxonomy design is fairly easy to grasp and follow for them. Furthermore, four out of five participants expect that it would be *easy* or *very easy* for them to *extend* the ontology (question **b**). The modeled hazard and safety

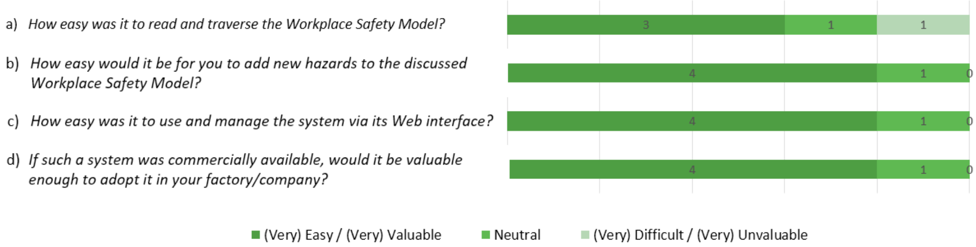


Fig. 6: The interviewed domain experts generally found it easy to use the system and believe that the adoption of a system like this would be valuable for their company.

concepts thus seem to be externally valid and it seems to be possible to extract these concepts from standardized safety recommendations for people who are not subject-matter experts (i.e., the authors). Furthermore, we could show that subject-matter experts with little experience with ontological modeling are indeed able to read the ontology, understand its contents, and extend parts of it without prior training – proper graphical ontology exploration tools should be suitable to further facilitate this task.

We also asked the survey participants about their opinion of our approach and implementation as a whole. The participants said that they find our concrete prototype system – in particular the dashboard interface – straightforward to use for configuring and controlling manufacturing processes (question **c**). In summary, they would wish to use such a system to facilitate the implementation of workplace safety regulation within their professional environment and believe that a system like this would be valuable to adopt for their company (question **d**). From our limited survey among domain experts, we thus conclude that our approach and system would be valuable in a realistic setting and that its design and set-up are in-line with experts' expectations regarding future industrial manufacturing automation systems.

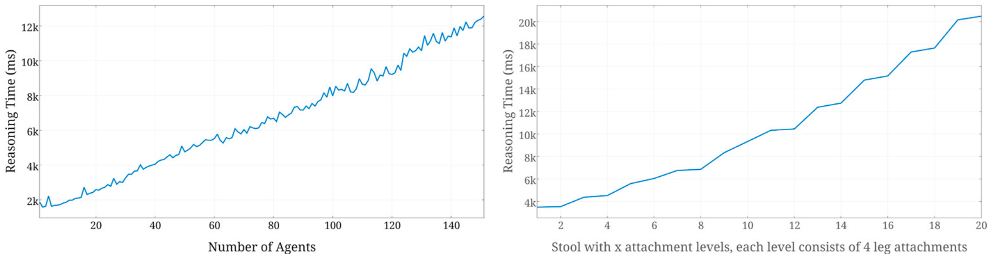
4.2. *Prototype Evaluation: Scalability*

In addition to the qualitative opinions about our approach, we quantitatively evaluated one major aspect of automated industrial planning systems: the scalability of our implementation given an increasing *Number of Agents* and with respect to the *Complexity of the Product* to be assembled. To do this, we performed several automatic tests and measured its reasoning time when generating increasingly complex collaborative assembly plans while taking into account workplace safety regulation.

First, we measured the planning time of our system with respect to an increasing *number of agents*. Every iteration of this test generated an assembly task plan for a 4-leg stool that consists of a seat with four attached legs. As can be seen in Figure 7a, the reasoning time was measured while increasing the number of agents to up to 150, causing the performance of the system to decrease linearly: it took our system 1.8s to generate the 4-leg stool assembly plan with only a single available agent, and 12.3s when 150 agents were available.[8] As our system's performance behaves linearly in this respect, reasoning time can be improved by scaling up the processing capabilities of the machine that hosts the *Semantic Server*.

Next, we measured the system's reasoning time while increasing the *complexity of the furniture product*: in each iteration of this test, we generated an assembly task plan for a stool that consists of a seat, and i attachment levels, where each attachment level consists of 4 legs connected to the previous attachment level via screws. We measured our system's reasoning time while increasing the number of attachment levels to up to 20 – the measured system performance with respect to increasing the complexity of the furniture product is shown in Figure 7b. The system needed 3.4s to generate an assembly plan for a stool with a single attachment level, 9.3s for a stool with 10 attachment levels (i.e., 1 million legs in total), and 20.4s for a stool with 20 attachment levels (i.e., 1 trillion legs in total). This

[8] These tests were carried out on an i5-6500T CPU with 2.5GHz and 8GB of RAM



(a) (b)

Fig. 7: Scalability evaluation of our system: (a) Reasoning time of our system with increasing number of (simulated) available agents. (b) Reasoning time of our system with increasing product complexity.

almost performance is due to the assembly task planning rules being executed in parallel for every leg – each additional attachment *level* of the product incurs a single additional reasoner iteration however, yielding the measured behavior.

Our system's performance both with respect to the number of involved agents and the complexity of the products is sufficient for productively applying it in the context of the automatic planning of medium-sized manufacturing processes (e.g., in SMEs) as well as to individual portions of large manufacturing processes (e.g., those with human-robot interactions, or those that require a high degree of flexibility). In particular, integrating the automatic enforcement of workplace safety regulation did not have an large adverse effect on the performance of the system: the measured performance numbers put our approach in-between mechanisms that rely on goal-decomposition trees and achieve linear scaling with faster overall performance such as [17], and highly flexible first-order logic based composition systems that scale exponentially such as [2, 18].

5. Conclusions

The digitalization of industries and semantic coupling of services, products, machines, and people will give rise to unprecedented possibilities for optimizing manufacturing processes and making them more flexible. In this work, we focus on the automatic enforcement of workplace safety regulation in goal-driven industrial manufacturing processes: we propose a system that is capable of creating collaborative manufacturing plans for human and machine agents while taking into account individual as well as generic constraints of the agents. Generated plans account for workplace safety regulation by considering potential hazards and integrating mitigation strategies – this is done by combining a semantic reasoner with knowledge models that codify complex workplace safety constraints published by the U.S. OSHA while integrating real-time data from a visual tracking device. We were able to demonstrate that our prototype is scalable both with respect to the number of agents and product complexity, and – importantly – that the system's underlying semantic models can be understood and extended by subject-matter experts with only negligible prior training. This aspect is crucial as we believe that one major obstacle toward the broader adoption of information models of this kind lies in the process of *translating* from human-readable documents to machine-readable ontologies and in *managing* the resulting models – ideally, regulatory bodies such as the U.S. OSHA or the German Bundesamt für Arbeitsschutz und Arbeitsmedizin (BAuA) would already publish workplace safety rules in a format can readily be integrated with industrial automation software.

Using semantics to insert cues about safe practices during factory operations has the potential to save life and limb. Our approach and prototype system contribute to avoiding work-related injuries while at the same time reducing regulatory cost for manufacturers. The same principle can be used to enforce more general constraints for assembly planning systems, and beyond: since our approach treats human and robotic agents identically on an abstract level and differences between the agents are manifest only on lower levels (e.g., an agent's anatomy), we are able to incorporate a broad range of constraints, including equipment requirements, spatial constraints for agents, and timing constraints (e.g., when reasoning about industrial mixing in the chemical industry).

References

[1] V. Issarny, N. Georgantas, S. Hachem, A. Zarras, P. Vassiliadist, M. Autili, M. A. Gerosa, A. B. Hamida, Service-oriented middleware for the future internet: State of the art and research directions, JISA 2 (1) (2011) 23–45.
[2] S. Mayer, D. Plangger, F. Michahelles, S. Rothfuss, UberManufacturing: A goal-driven collaborative industrial manufacturing marketplace, in: Proc. IoT, 2016.
[3] S. Zhang, F. Boukamp, J. Teizer, Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for Job Hazard Analysis (jha), Automation in Construction 52 (2015) 29–41.
[4] R. Schank, R. Abelson, Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures, Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1977.
[5] M. Grüninger, C. Menzel, The Process Specification Language (PSL) theory and applications, AI Magazine 24 (3) (2003) 63–74.
URL http://dl.acm.org/citation.cfm?id=958671.958677
[6] Instrumentation, Systems, and Automation Society, ANSI/ISA-88.01-1995 Batch Control Part 1: Models and Terminology, from website (1995).
URL http://www.automatedresults.com/documents/S88-Batch-Standard.pdf
[7] D. Kortenkamp, D. Schreckenghost, Developing and executing goal-based, adjustably autonomous procedures, in: Proc. AIAA I@A, 2007.
[8] M. Tenorth, M. Beetz, KnowRob – A knowledge processing infrastructure for cognition-enabled robots, Int. Journal of Robotics Research 32 (5) (2013) 566 – 590.
[9] S. Saad, D. D. Beul, S. Mahmoudi, P. Manneback, An ontology for video human movement representation based on Benesh notation, in: Proc. ICMCS, 2012.
[10] M. P. Reed, B.-K. Park, K. H. Kim, U. Raschke, Creating custom human avatars for ergonomic analysis using depth cameras, in: Proc. HFES, 2014.
[11] H.-P. Wiendahl, J. Reichardt, P. Nyhuis, Handbook Factory Planning and Design, Springer, 2015.
[12] A. M. Bestick, S. A. Burden, G. Willits, N. Naikal, S. S. Sastry, R. Bajcsy, Personalized kinematics for human-robot collaborative manipulation, in: Proc. IROS, 2015.
[13] M. Beetz, L. Mösenlechner, M. Tenorth, CRAM – A Cognitive Robot Abstract Machine for everyday manipulation in human environments, in: Proc. IROS, 2010.
[14] S. Mayer, J. Hodges, D. Yu, M. Kritzler, F. Michahelles, An open semantic framework for the industrial Internet of Things, IEEE Intelligent Systems 32 (1) (2017) 96–101.
[15] S. Mayer, E. Wilde, F. Michahelles, A connective fabric for bridging Internet of Things silos, in: Proc. IoT, 2015.
[16] J. Hodges, Naive mechanics: A computational model for representing and reasoning about simple mechanical devices (December 1993).
[17] A. Ciortea, O. Boissier, A. Zimmermann, A. M. Florea, Responsive decentralized composition of service mashups for the internet of things, in: Proc. IoT, 2016.
[18] M. Kovatsch, Y. N. Hassan, S. Mayer, Practical semantics for the Internet of Things: Physical states, device mashups, and open questions, in: Proc. IoT, 2015.