Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 93 (2020) 1097–1102

www.elsevier.com/locate/procedia

53rd CIRP Conference on Manufacturing Systems

An Ontology for Human-Robot Collaboration

Alessandro Umbrico, Andrea Orlandini*, Amedeo Cesta

CNR - Institute of Cognitive Science and Technology (CNR-ISTC), Via S. Martino della Battaglia 44, Rome 00185, Italy

Abstract

The diffusion of Human-Robot Collaborative cells is prevented by some barriers. Classical control approaches seem not yet fully suitable for facing variability conveyed by the presence of human operators beside robots. Heterogeneous knowledge representation capabilities and abstract reasoning are crucial to enhance flexibility of control solutions. This work presents SOHO (*Sharework Ontology for Human Robot Collaboration*), a novel ontology specifically designed for Human-Robot Collaboration. The paper describes the pursued context-based approach, the novelty of the designed ontology with respect to the state of the art and shows its validity in a realistic Human-Robot Collaboration scenario.

© 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 53rd CIRP Conference on Manufacturing Systems

Keywords: Human-Robot Collaboration; Artificial Intelligence; Formal Ontology;

1. Introduction

Nowadays, robots are successfully deployed in a large spectrum of real-world applications. Nevertheless, research activities are still ongoing to enable robots to autonomously operate in environments, i.e., understanding the actual situation, planning their tasks and acting to safely and effectively achieve some given goals. Classical control processes usually rely on static models of the working cells that may become obsolete while operating, especially in dynamic contexts. In Human-Robot Collaboration (HRC), robots share the working space and tightly interact with humans whose behaviors are neither predictable nor controllable. Dealing with HRC scenarios entails integrating Artificial Intelligence and Robotics solutions enabling collaborative robots to i) have knowledge about the environment (and "interpret" it), ii) reason over its possible actions to modify the environment and iii) act in the environment to achieve a desired objective [1].

We are investigating the enrichment of collaborative robot control systems through a perceive-reason-act paradigm implementing advanced cognitive features. Such features would allow a robot to perceive the production environment, recognize events and activities and dynamically adapt its behavior accordingly. From a perception perspective, semantic technologies are well suited to realize the representation and reasoning capabilities necessary to achieve our objectives. From a control perspective, the integration of knowledge-based technologies is crucial to increase the flexibility of robot behaviors when interacting with humans [2–4] or dealing with changing production needs as in Reconfigurable Manufacturing Systems [5, 6].

Semantic technologies are crucial to uniformly interpret and represent heterogeneous information coming from different sensing devices. The envisaged cognitive processes should rely on a well defined ontology in order to fuse and contextualize information extracted from different sensory sources and infer increasingly abstract knowledge about the production environment (e.g., the configurations of a production line, the capabilities of robots, tasks being performed by human operators, etc). Our contribution takes inspiration from Sharework (https://sharework-project.eu/), a H2020 research project funded by the European Commission within "Factories of the Future" programme whose aim is to design and demonstrate an advanced control solutions for coordinating human workers and collaborative robots in a reliable, safe and efficient way.

This work presents an overview of the ontology, SOHO (*Sharework Ontology for Human Robot Collaboration*), describing the pursued modeling approach, the novelty of SOHO with respect to existing ontologies in robotics and the key novel aspects and modeling features.

* Corresponding author. Tel.: +39-06-44595-223; fax: +39-06-44595-243.
E-mail address: andrea.orlandini@istc.cnr.it (Andrea Orlandini).

2212-8271 © 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 53rd CIRP Conference on Manufacturing Systems
10.1016/j.procir.2020.04.045

2. Ontology in Computer Science and Robotics

Ontologies can be seen as formal descriptions of objects, properties and relationships among objects collected in a particular data structure called the Knowledge Base (KB). In computer science, ontologies have been defined in different ways by different scientists. Studer et al. [7] combined the definitions by Gruber [8] and Borst [9] stating that an ontology is "an explicit, formal specification of a shared conceptualization. The different characterizations of ontology are complementary and can be combined together. Hence, according to [10], it can be said that "an ontology is an artificial representation, that represents types or universals of a certain domain and the relations that hold according to a certain theory in a formal structure.

Depending on the specific application needs, different types of ontology can be defined. Guarino proposes a classification based on levels of generality [11] and defines four classes: (i) *Top-level or Upper ontologies* describe very general concepts like e.g., space, time, event or action, that are independent from a particular problem or domain; (ii) *Domain ontologies* describe general concepts related to a specific domain; (iii) *Task ontologies* describe generic tasks or activities. (iv) *Application ontologies* characterize a specific application and describe concepts whose relevance is limited to a specific domain and task.

Our aim is to propose a novel *domain ontology* and formally define concepts, properties and relationships that are relevant in collaborative manufacturing scenarios. Many existing ontologies like, e.g., [12, 13], characterize cyber-physical production systems as a whole or specific production aspects but do not take into account *collaborative aspects*. Two contributions we found relevant to our objectives are: (i) the CORA ontology [14] and; (ii) the SSN ontology [15].

CORA is an IEEE standard ontology for robotics and automation. It has been defined with the aim of promoting a common language in the robotics and automation domain. It proposes a semantics to formally characterize knowledge about robots and robot parts, robot positions and configurations and groups of robots. This standard relies on SUMO as theoretical foundation and integrates the framework ALFUS [16] to characterize the autonomy levels of a robot.

SSN is a W3C standard ontology for IoT devices and sensor network. It defines basic concepts and properties characterizing the capabilities of sensing devices, their deployment into a physical environment and the physical phenomena such devices can observe. SSN relies on DUL (a subset of DOLCE) as theoretical background. It extends abstract concepts like e.g., `DUL:Quality` and `DUL:Region` to represent respectively physical properties that can be observed and metics that can be used to measure the outcome of sensing processes.

2.1. Foundations

Upper ontologies aim at describing reality from a quite general perspective in order to define very general concepts that are the same across all domains. The concepts and the properties defined by upper ontologies may seem quite abstract and not much useful in concrete problems but the use of this kind of ontologies (also known as foundational ontologies) is quite recommended [10, 17].

The use of upper ontologies represents a good design choice to build new domain ontologies. As shown in [10], these concepts represent a stable theoretical foundation fostering a clear structuring and disambiguation of new concepts and related relationships. According to [17], upper ontologies indeed guide a correct classification of knowledge entities of a particular domain and facilitate interoperability among different ontologies.

A number of upper ontologies exist in the literature. SUMO, DOLCE and BFO are some examples of the most famous and used. Each upper ontology has its own basic assumptions that characterize general and abstract concepts. For example, DOLCE is an "ontology of particulars. It does have universal (classes and properties), but the claim is that they are only employed in the service of describing particulars. In contrast, SUMO could be described as an ontology of both particulars and universals. Also, DOLCE uses meta-properties as a guiding methodology, while SUMO pursues a formal definition of such meta-properties directly in the ontology itself (axiomatization). The work [17] gives a first comparison of these and other upper ontologies known in the literature.

Although similar, these ontologies cannot be directly integrated without introducing contradictions. Some works have focused on the definition of the so called upper-upper ontologies for the integration of different foundational ontologies [18]. Since CORA and SSN rely on two different upper ontologies and since upper-upper ontologies like e.g., COSMO (COmmon Semantic MOdel) have obtained poor practical results, a first design choice was the selection of an upper ontology for SOHO.

2.2. What is Missing for HRC?

CORA and SSN are well structured ontologies defining concepts and properties that are relevant for HRC but they do not cover all the necessary information. The scope of SSN is limited to the characterization of a physical environment in terms of properties that can be observed and sensing devices that carry out "sensing processes". This ontology is quite "self-contained" and can be easily integrated with CORA to represent also robot interfaces and sensing parts. CORA has a broader scope. It focuses on robot parts, robot configurations and levels of autonomy. However, CORA does not support the contextualization and interpretation of behaviors of robots and other autonomous agents (e.g., human operators) with respect to the global production objectives and processes. For example, CORA does not consider the Human as an autonomous agent operating in autonomy or in collaboration with robots to achieve a common (production) objective. More specifically, three main limitations can be pointed out.

Functions, Tasks and Capabilities. A detailed description of production processes and tasks that *agents* (e.g., human workers or robots) can perform is necessary to dynamically coordinate the available resources. Such a structured description is crucial to realize a flexible collaboration between humans and robots.

Humans as collaborative agents. A detailed description of human operators in terms of capabilities and their "autonomy level" is crucial to dynamically adapt and coordinate collaborative processes. In this regard, it would be interesting to extend the ALFUS model to human workers. This knowledge together with a model of possible collaboration modalities of tasks is necessary to reason about safety requirements and synthesize collaborative plans accordingly.

Intentions, commitment and coordination issues. The envisaged system should be capable of recognizing human behaviors from sensor observations and contextualize them with respect to production objectives. It is necessary to represent and reason about abstract concepts like e.g., *human intentions* and link observed behaviors to (known) production processes in order to react or adapt planned operations accordingly.

To overcome these limitations we here present SOHO as a novel domain ontology built on top of CORA and SSN. We rely on DOLCE as foundational layer to follow its endurant interpretation of some relevant concepts and thus define a (domain) ontology with a high level of flexibility. While SSN already relies on DOLCE, it is required an extra efforts to revise and adapt the definition of some key concepts of CORA which relies on SUMO.

3. Design of a Domain Ontology for HRC

According to [10], ontologies have to be adequate to their domains, and domains come along on different granularity levels. Therefore, an ontology needs to account for all levels that are relevant for a domain. To pursue this general principle we follow a context-based approach and organize SOHO in a number of contexts: (i) the *Environment Context*; (ii) the *Behavior Context* and; (iii) the *Production Context*. Each context describes a target domain from a particular perspective and granularity level.

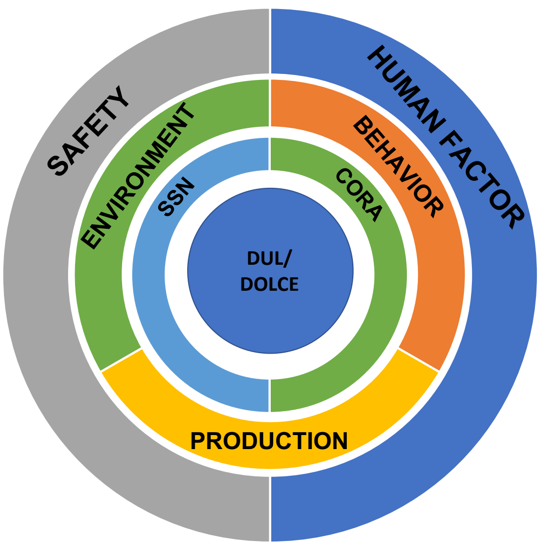

Fig. 1. Conceptual view of SOHO

Figure 1 shows a conceptual view of these contexts and the overall organization of SOHO. In particular, it points out the correlations between the three defined contexts and the underlying extended ontologies.

The *safety* and *human factor* "contexts" of Figure 1 emphasize the HRC perspective of the defined contexts. They do not represent actual ontological contexts. Rather they define two "meta-perspectives" that must be uniformly considered by all ontological contexts. Defined concepts and properties should thus take into account safety issues and human-factors in order to support reasoning processes for the synthesis of *safe collaborations* between humans and robots in dynamic scenarios.

3.1. Environment Context

The environment context defines physical elements and general properties of an environment that can be observed. This context strongly relies on SSN which is crucial to characterize the sensing capabilities of available devices and the physical properties of domain entities they can observe.

We extend SSN in order to uniformly represent physical objects that can be part of a production environment, their properties like e.g., position in space, orientation, weight, etc., and the kind of data that can be gathered through their observation. To characterize the *observable features* of an environment we pursue an interpretation more flexible than the one proposed by SSN. We rely on the concept of `DUL:Role` to avoid a rigid classification of `SSN:ObservableProperty` as subclass of `DUL:Property` and therefore as some intrinsic quality of an object that can be observed.

We define observable properties as a `DUL:Role` that a property of an object can play according to the sensing capabilities of the "environment. While a `DUL:Property` represents an intrinsic quality of an object the "observability of such a property depends on the sensing capabilities of available devices. The ontology thus defines the concept `ObservableFeature` as a `DUL:Role` that objects can play according to the sensing capabilities of the deployed sensors. On the one hand, an object is characterized by a number of properties that reflect its own features and "nature and these properties are independent from the particular "application context. On the other, only a proper subset of these properties can be observed, depending on the sensing capabilities of a particular "application scenario.

$$\begin{aligned}\mathtt{ObservableFeature} \sqsubseteq\ & \mathtt{DUL{:}Role} \sqcap \mathtt{SSN{:}FeatureOfInterest} \sqcap \\ & \exists\, \mathtt{DUL{:}isRoleOf.ProductionObject} \sqcap \\ & \exists\, \mathtt{hasObservableProperty.ProductionProperty} \sqcap \\ & \exists\, \mathtt{isObservableThrough.SSN{:}Sensor}\end{aligned}$$

Our choices of defining `ObservableFeature` and the property `hasObservableProperty` emphasizes the decoupling between the intrinsic properties of an object and the ones that can actually be observed. This distinction allows us to contextualize the perception capabilities of a production scenario according to the deployed sensing devices as shown by the rule below.

$$\begin{aligned}& \mathtt{ProductionObject(o)} \land \\ & \mathtt{DUL{:}hasProperty(o,\ p)} \land \\ & \mathtt{SSN{:}Sensor(s)} \land \\ & \mathtt{SSN{:}observes(s,\ p)} \\ & \quad \rightarrow \mathtt{ObservableFeature(x)} \land \\ & \quad\quad \mathtt{DUL{:}isRoleOf(x,\ o)} \land \\ & \quad\quad \mathtt{hasObservableProperty(x,\ p)} \land \\ & \quad\quad \mathtt{isObservableTrhough(x,\ s)}\end{aligned}$$

For example, human tracking processes can be realized only if the considered working environment is endowed with sensing devices capable of observing physical properties of a human worker like e.g., physical position, posture, motion speed and direction, etc.

3.2. Behavior Context

The Behavior Context defines capabilities and tasks agents can perform. It integrates the Taxonomy of Functions defined in [19] which characterizes the tasks or low-level operations that manufacturing agents can perform. In our case, agents can be either robots or human workers. The taxonomy characterizes low-level operations (*primitive tasks*) as `Function` and classifies them according to the effects they have on the `DUL:Quality` of production objects.

$$\begin{aligned}\texttt{Function} \sqsubseteq\ & \texttt{DUL:Method} \sqcap \\ & \exists\, \texttt{DUL:isDescribedBy.ExecutionNorm} \sqcap \\ & \exists\, \texttt{hasEffectOn.DUL:Quality} \sqcap \\ & (\exists\, \texttt{requires.Capability} \sqcup \\ & \exists\, \texttt{requires.ProductionObject})\end{aligned}$$

Functions are defined in terms of the capabilities needed to perform them. The concept `Capability` defines general interaction primitives of humans and robots and can be further distinguished between `ActingCapability` and `SensingCapability`.

Also, this context characterizes human and robot behaviors by defining the different the levels of collaboration that may characterize task execution.. Specifically, we take into account the framework ALFUS [16] to represent the levels of autonomy of robots and extend this model to human operators in order to represent their ability of working in autonomy. Workers with different expertise and backgrounds indeed may have different skills and different levels of autonomy.

Each `Function` is associated with an `ExecutionNorm` which describes the way a function is carried out with respect to some `InteractionModality`. An `InteractionModality` is a `DUL:Quality` of an execution norm and characterizes the way a `Function` should be executed in a collaborative scenario. Taking inspiration from the classification proposed in [20], we define four types of interaction modalities: (i) `Synchronous`; (ii) `Simultaneous`; (iii) `Supportive`; (iv) `Independent`.

As a `DUL:Quality`, an `InteractionModality` can be "measured by means of some `DUL:Region` that define dimensional spaces that can be used as a value for measuring the quality of an entity. In this context, a particularly interesting aspect to measure is the *risk* of an interaction modality with respect to the safety of a human. To this aim, we define the concept of `RiskLevel` as a specialization of the class `DUL:Region` to measure the risk entailed by a particular interaction modality in a particular collaboration scenario.

3.3. Production Context

The production context strongly relies on DUL and defines concepts and properties that characterize production processes in terms of objectives and operations humans and robots should perform to successfully achieve them. A proper representation of this knowledge is crucial to support human-robot coordination processes and establish human and robot *commitment* to production goals [21, 22] and a level of *agreement* about the way they achieve these goals [23, 24]. The definition of necessary concepts rely on the foundational concepts `DUL:Event` and `DUL:Description` in order to distinguish between the general description of the possible ways of carrying out a process and its particular implementations.

The description of a production process follows a task-oriented approach. Tasks characterize both the structure of production processes and the possible collaborations as tasks can be associated and allocated to different resources like e.g., human workers, robots or machines, according to their availability and production needs. This approach follows the hierarchical organization proposed by [25, 26] to characterize production goals and processes (Hierarchical Task Analysis).

The main element is the *goal* which defines general objectives that must be achieved through the execution of a number of operations (i.e., production plans). We first define the concept `ProductionGoal` as a specialization of `DUL:Goal` to characterize general objectives of a production plant. Each `ProductionGoal` is associated with a number of `ProductionMethod` (at least one method for each goal is necessary) and can be described by a `ProductionPlan`.

A `ProductionPlan` describes a particular implementation of a particular `ProductionMethod` in terms of actions whose execution satisfies a particular `ProductionGoal`.

$$\begin{aligned}\texttt{ProductionPlan} \sqsubseteq\ & \texttt{DUL:Plan} \sqcap \\ & \exists\, \texttt{DUL:isParticipantIn.ProductionProcess} \sqcap \\ & \exists\, \texttt{DUL:describes.ProductionGoal} \sqcap \\ & \exists\, \texttt{SSN:implements.ProductionMethod}\end{aligned}$$

A `ProductionMethod` always refers to one particular `ProductionGoal`. Vice versa a `ProductionGoal` can be associated with a number of `ProductionMethod`. Each `ProductionMethod` is composed by a number of `ProductionTask` and is associated with a number of `ProductionRelation` specifying "constraints" about task execution. The property `DUL:hasConstituent` emphasizes the hierarchical composition of a `ProductionMethod` as made of a number of `ProductionTask` that could be complex and hierarchically organized as well.

$$\begin{aligned}\texttt{ProductionMethod} \sqsubseteq\ & \texttt{DUL:Method} \sqcap \\ & \exists\, \texttt{DUL:hasComponent.ProductionRelation} \sqcap \\ & \exists\, \texttt{DUL:hasConstituent.ProductionTask} \sqcap \\ & \exists!\, \texttt{isRelatedTo.ProductionGoal}\end{aligned}$$

4. Capabilities, Processes and Collaborative Plans

The concept `ProductionMethod` is central to the description of a production process and is defined as the composition of a number of production tasks. We define three types of `ProductionTask`: (i) `ComplexTask`; (ii) `SimpleTask`; (iii) `Function`.

A `ComplexTask` is a `DUL:Method` used to characterize the hierarchical structure of a process linking complex tasks to increasingly simpler ones. The hierarchical structure is enforced by the property DUL:hasConstituent which associates ComplexTask with either SimpleTask or other ComplexTask. This enables the representation of tasks with arbitrary complex hierarchical structures.

$$\begin{aligned}\texttt{ComplexTask} \sqsubseteq\ & \texttt{DUL:Method} \sqcap \\ & (\exists\, \texttt{DUL:hasConstituent.SimpleTask} \sqcup \\ & \exists\, \texttt{DUL:hasConstituent.ComplexTask}) \sqcap \\ & \exists\, \texttt{requires.ProductionObject}\end{aligned}$$

$$\begin{aligned}\texttt{SimpleTask} \sqsubseteq\ & \texttt{DUL:Method} \sqcap \\ & \exists\, \texttt{DUL:isDescribedBy.ExecutionNorm} \sqcap \\ & \exists\, \texttt{DUL:hasConstituent.Function} \sqcap \\ & (\exists\, \texttt{requires.Capability} \sqcup \\ & \exists\, \texttt{requires.ProductionObject})\end{aligned}$$

A SimpleTask represents a *leaf* of a task hierarchy and therefore describes a *primitive production operation* that can be carried out through the execution of some Function. Depending on the specific needs of a production scenario, such simple tasks and functions should be executed following some collaboration requirement. The concept ExecutionNorm models such requirements and may represent particular InteractionModality affecting the execution tasks.

The separation between interaction capabilities and functions and the more general structure of production processes are crucial to support a flexible allocation of tasks and consequently a dynamic synthesis of collaborative plans.

This semantics support knowledge reasoning mechanisms that can dynamically reason on the capabilities of agents and analyze possible collaborations. To this aim, we define inference rules that support such knowledge processing mechanisms and contextualize agents' capabilities with respect to the requirements of a particular production process.

$$\begin{aligned}&\texttt{DUL:Agent(a)} \wedge \texttt{Function(f)} \wedge \\ &\texttt{hasCapability(a, c)} \wedge \texttt{requires(f, c)} \\ &\qquad \rightarrow \quad \texttt{canPerform(a, f)}\end{aligned}$$

$$\begin{aligned}&\texttt{DUL:Agent(a)} \wedge \texttt{Function(f)} \wedge \\ &\texttt{canPerform(a, f)} \wedge \texttt{SimpleTask(t)} \\ &\texttt{requires(t, f)} \rightarrow \quad \texttt{canBeAssignedTo(t, a)}\end{aligned}$$

The rules above contextualize respectively the functions an agent can perform according to its capabilities and, possible task assignments according to the functions that agents can actually perform. Knowledge inferred through the interpretation of these rules is a basis for carrying out commitments between humans and robots that enable a flexible assignment of tasks.

Given this general organization, the actual execution of a collaborative process is achieved following the instructions encapsulated by some ProductionPlan which implements a particular ProductionMethod. The *event of executing a production plan* is then represented within the class DUL:Event. We have defined the concept CollaborativeProcess as a specialization of DUL:Process in order to represent production events that *modify over time* the state of the production environment from an initial DUL:Situation to a final/resulting DUL:Situation.

$$\begin{aligned}\texttt{CollaborativeProcess} \sqsubseteq\ & \texttt{DUL:Process} \sqcap \\ & \exists\, \texttt{hasInitialSetting.DUL:Situation} \sqcap \\ & \exists\, \texttt{hasResultingSetting.DUL:Situation} \sqcap \\ & \exists\, \texttt{DUL:isDescribedBy.ProductionPlan} \sqcap \\ & \exists\, \texttt{DUL:hasParticipant.Robot} \sqcap \\ & \exists\, \texttt{DUL:hasParticipant.HumanWorker} \sqcap \\ & \exists\, \texttt{DUL:hasPart.ProductionRelatedEvent} \sqcap \\ & \exists\, \texttt{DUL:hasPart.ProductionAction}\end{aligned}$$

Similarly to production processes, production actions are defined within the concept DUL:Event. A ProductionAction is indeed defined as a specialization of DUL:Action and therefore as an event associated to a Schedule specifying its (flexible) temporal occurrence and where at least one DUL:Agent participates in, according to some InteractionModality.

$$\begin{aligned}\texttt{ProductionAction} \sqsubseteq\ & \texttt{DUL:Action} \sqcap \\ & \exists\, \texttt{DUL:hasParticipant.DUL:Agent} \sqcap \\ & \exists\, \texttt{DUL:hasParticipant.ProductionObject} \sqcap \\ & \exists\, \texttt{DUL:hasProperty.InteractionModality} \sqcap \\ & \exists\, \texttt{DUL:hasProperty.Schedule} \sqcap \\ & \exists\, \texttt{DUL:hasPart.ProductionRelatedEvent}\end{aligned}$$

Then, the class ProductionAction is further expanded by taking into account the classification of interactions proposed by CORA. We define a number of robot and human actions and a number of possible human-robot interactions to respectively characterize actions that agents carry out autonomously and actions that require some kind of interactions.

4.1. A Realistic Collaborative Scenario

Let us consider an assembly scenario where a robotic arm and a worker collaborate to build a particular work piece on a shared working table. SOHO (TBox) defines all the concepts and properties needed to properly model the structure and the requirements of such a production process.

ProductionGoal represents the high-level collaborative process to perform. ProductionMethod defines the set of simple and complex tasks the human and the robot should perform to successfully carry out the considered process. ComplexTask describes complex assembly activities (e.g., "assemble the top cover") representing sets of operations to be performed according to a specific order (e.g., "assemble the bottom cover before the top cover"). Such ordering constraints are described by means of a number of ProductionRelation. SimpleTask describes a "primitive operations" composing a complex task (e.g., "fix part *x*", "do quality inspection" or "move object *x* from location *y* to location *z*"). Such tasks require the execution of a number of Functions the robot and/or the human can perform according to their capabilities. Examples of such functions are "*screw* bolt *x*" or "*pick* object *y* from location *z*". Simple tasks and functions could be performed according to a specific InteractionModality. For example, simple tasks/functions concerning the movement of heavy objects may require physical interactions between a human and a robot and thus a *Supportive* interaction modality. Others tasks/functions like e.g., "unscrewing bolts", can be performed autonomously by human or robot as *Independent* interactions.

An *instantiation* of SOHO can be leveraged by the control system of a collaborative robot to build an internal *knowledge* (ABox) and achieve an higher level of *awareness* of the production context enabling safer and more flexible collaborative behaviors. Indeed, considering a *reactive approach*, such knowledge would allow to, e.g., dynamically *recognize* which is the task performed by a human, to *infer* the production process being executed and then *dynamically select* the most suited robot task for supporting production. Rather, considering a *deliberative approach*, given an execution request for a collaborative assembly process, the knowledge allows to reason over the production process as well as over robot and operator capabilities in order to autonomously synthesize a flexible *collaborative plan*. Such a plan would define tasks and functions to be performed by human and robot to satisfy a requested production goal (i.e., a `ProductionPlan` implementing a particular `ProductionMethod` achieving the requested `ProductionGoal`).

5. Conclusions and Future Works

This paper presents SOHO (*Sharework Ontology for Human Robot Collaboration*), a novel domain ontology for Human-Robot Collaboration defined within the Sharework H2020 research project. SOHO formally characterizes HRC manufacturing scenarios by considering different perspectives. Its main original feature indeed relies on the use of a context-based approach to ontology design, supporting flexible representation of collaborative production processes. Future research directions will focus on extensions of SOHO and its deployment in realistic collaborative industrial scenarios. On one hand, we aim at further investigating safety aspects concerning physical Human-Robot interactions to better characterize safety properties of robotic devices. On the other hand, effective deployment requires limiting latencies introduced by the use of such semantic technologies with potential overhead due to, e.g., possible inconsistencies in a knowledge base. A good tradeoff between the granularity level of ontological models and the scope of resulting knowledge reasoning processes is crucial to balance performance of robot control and behavior flexibility and thus effectively deploy such technologies to real production environments Finally, our long-term research objective is to foster the use of SOHO and integrate knowledge reasoning mechanisms into cognitive AI-based controllers for collaborative working cells so as to enable more flexible and "natural" collaboration between human workers and robots.

Acknowledgement

Authors are partially supported by the European Commission within the Sharework project (H2020 Factories of the Future GA No. 820807).

References

[1] K. Rajan, A. Saffiotti, Towards a science of integrated ai and robotics, Artificial Intelligence 247 (2017) 1 – 9.

[2] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, R. Alami, Artificial cognition for social human-robot interaction: An implementation, Artificial Intelligence 247 (2017) 45 – 69, special Issue on AI and Robotics.

[3] M. Tenorth, M. Beetz, Representations for robot knowledge in the KnowRob framework, Artificial Intelligence.

[4] I. Awaad, G. K. Kraetzschmar, J. Hertzberg, The Role of Functional Affordances in Socializing Robots, International Journal of Social Robotics 7 (4) (2015) 421–438.

[5] S. Borgo, A. Cesta, A. Orlandini, A. Umbrico, Knowledge-based Adaptive Agents for Manufacturing Domains, Engineering with Computers 35 (3) (2019) 755–779.

[6] S. Borgo, A. Cesta, A. Orlandini, A. Umbrico, A planning-based architecture for a reconfigurable manufacturing system, in: The 26th International Conference on Automated Planning and Scheduling (ICAPS), 2016, pp. 358–366.

[7] R. Studer, V. R. Benjamins, D. Fensel, Knowledge engineering: Principles and methods, Data & Knowledge Engineering 25 (1) (1998) 161 – 197.

[8] T. R. Gruber, Toward principles for the design of ontologies used for knowledge sharing?, International Journal of Human-Computer Studies 43 (5) (1995) 907 – 928.

[9] B. Willem Nico, Construction of engineering ontologies for knowledge sharing and reuse, Ph.D. thesis, Centre for Telematics and Information Technology (CTIT) (1997).

[10] L. Jansen, S. Schulz, The ten commandments of ontological engineering, in: Proceedings of the 3rd Workshop of Ontologies in Biomedicine and Life Sciences, 2011, pp. 1–6.

[11] N. Guarino, Formal Ontology in Information Systems, IOS Press, 1998.

[12] E. M. Sanfilippo, Y. Kitamura, R. I. M. Young, Formal ontologies in manufacturing, Applied Ontology 14 (2019) 119–125.

[13] Z. Usman, R. Young, N. Chungoora, C. Palmer, K. Case, J. Harding, Towards a formal manufacturing reference ontology, International Journal of Production Research 51 (22) (2013) 6553–6572.

[14] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, A. Chibani, S. Grard, Y. Amirat, C. Schlenoff, Towards a core ontology for robotics and automation, Robotics and Autonomous Systems 61 (11) (2013) 1193 – 1204.

[15] M. Compton, P. Barnaghi, L. Bermudez, R. Garca-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, K. Taylor, The ssn ontology of the w3c semantic sensor network incubator group, Journal of Web Semantics 17 (2012) 25 – 32.

[16] H.-M. Huang, K. Pavek, B. Novak, J. S. Albus, E. R. Messina, A framework for autonomy levels for unmanned systems (alfus), in: Association of the Unmanned Vehicle System International (AUVSI), 2005.

[17] V. Mascardi, V. Cord, P. Rosso, A comparison of upper ontologies, in: Workshop from Objects to Agents (WOA), 2007, pp. 1–10.

[18] A. Kiryakov, K. I. Simov, M. Dimitrov, Ontomap: Portal for upper-level ontologies, in: Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001, Association for Computing Machinery, 2001, pp. 47–58.

[19] S. Borgo, M. Carrara, P. Garbacz, P. Vermaas, A formal ontological perspective on the behaviors and functions of technical artifacts, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 23 (1) (2009) 3–21.

[20] E. Helms, R. D. Schraft, M. Hagele, rob@work: Robot assistant in industrial environments, in: Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication, 2002, pp. 399–404.

[21] C. Castelfranchi, Commitments: From individual intentions to groups and organizations, in: Proceedings of the First International Conference on Multiagent Systems, 1995, pp. 41–48.

[22] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, R. Chatila, Toward human-aware robot task planning., in: AAAI spring symposium: to boldly go where no human-robot team has gone before, 2006, pp. 39–46.

[23] E. A. Sisbot, A. Clodic, L. F. Marin-Urias, M. Fontmarty, L. Brethes, R. Alami, Implementing a human-aware robot system, in: The 15th IEEE International Symposium on Robot and Human Interactive Communication, 2006, pp. 727–732.

[24] A. Clodic, R. Alami, R. Chatila, Key elements for human-robot joint action, in: Sociable Robots and the Future of Social Relations, 2014.

[25] J. A. Marvel, J. Falco, I. Marstio, Characterizing task-based human-robot collaboration safety in manufacturing, IEEE Transactions on Systems, Man, and Cybernetics: Systems 45 (2) (2015) 260–275.

[26] J. T. C. Tan, Feng Duan, Ye Zhang, Tamio Arai, Task decomposition of cell production assembly operation for man-machine collaboration by hta, in: 2008 IEEE International Conference on Automation and Logistics, 2008, pp. 1066–1071.