Article

RTMN 2.0—An Extension of Robot Task Modeling and Notation (RTMN) Focused on Human–Robot Collaboration

Congyu Zhang Sprenger [1,*], Juan Antonio Corrales Ramón [2,*] and Norman Urs Baier [1]

[1] School of Engineering and Computer Science, Institute of I3S, Bern University of Applied Science, 3400 Burgdorf, Switzerland; norman.baier@bfh.ch

[2] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain

* Correspondence: congyu.zhangsprenger@bfh.ch (C.Z.S.); juanantonio.corrales@usc.es (J.A.C.R.)

Abstract: This paper describes RTMN 2.0, an extension of the modeling language RTMN. RTMN combines process modeling and robot execution. Intuitive robot programming allows those without programming expertise to plan and control robots through easily understandable predefined modeling notations. These notations achieve no-code programming and serve as templates for users to create their processes via drag-and-drop functions with graphical representations. The design of the graphical user interface is based on a user survey and gaps identified in the literature We validate our survey through the most influential technology acceptance models, with two major factors: the perceived ease of use and perceived usefulness. While RTMN focuses on the ease of use and flexibility of robot programming by providing an intuitive modeling language, RTMN 2.0 concentrates on human–robot collaboration (HRC), which represents the current trend of the industry shift from "mass-production" to "mass-customization". The biggest contribution that RTMN 2.0 makes is the creation of synergy between HRC modes (based on ISO standards) and HRC task types in the literature. They are modeled as five different HRC task notations: Coexistence Fence, Sequential Cooperation SMS, Teaching HG, Parallel Cooperation SSM, and Collaboration PFL. Both collaboration and safety criteria are defined for each notation. While traditional isolated robot systems in "mass-production" environments provide high payload capabilities and repeatability, they suffer from limited flexibility and dexterity in order to be adapted to the variability of customized products. Therefore, human–robot collaboration is a suitable arrangement to leverage the unique capabilities of both humans and robots for increased efficiency and quality in the new "mass-customization" industrial environments. HRC has made a great impact on the robotic industry: it leads to increased efficiency, reduced costs, and improved productivity, which can be adopted to make up for the skill gap of a shortage of workers in the manufacturing industry. The extension of RTMN 2.0 includes the following notations: HRC tasks, requirements, Key Performance Indicators (KPIs), condition checks and decision making, join/split, and data association. With these additional elements, RTMN 2.0 meets the full range of criteria for agile manufacturing—light-out manufacturing is a manufacturing philosophy that does not rely on human labor.

Keywords: RTMN; HRC; HRC mode; HRC task; KPI; requirement; decision making

1. Introduction

Flexible robot programming is usually costly and requires much expertise. The development cycle often last months or years [1]. This puts significant pressure on operators and engineers. On the one hand, they are required to control the robot, and on the other hand, it is difficult to understand the robot's tasks due to a lack of high-level robotics expertise. Although frameworks and tools are available to simplify robot programming, these tools often are not flexible (brand-specific), are too technical, and are not intuitive enough for operators and engineers [1].

In our former research, we developed a modeling language called RTMN (Robot Task Modeling and Notation) [2]—an ontology-enabled, skill-based method of robot task modeling and notation. It aims to ease the modeling of the robotic process. The basic elements are based on Business Process Modeling and Notation (BPMN) [3]. The main motivation for developing RTMN was the challenges that non-robotics experts are facing in the fast-growing agile production industry: cost and requirement of robot programming expertise, as well as knowledge of the complexity and difficulties of controlling robotic systems. In order to bridge this gap, the authors introduced a model-driven framework that allows for intuitive modeling and programming of robotic processes using RTMN—an ontology-enabled skill-based robot task model and notation—that enables non-experts to plan and program robot tasks. In this research work, the authors conducted a literature analysis, adopting both quantitative (questionnaires) and qualitative (interviews) research methods, which balanced the limitations of one type of method with the strengths of another and enabled a more comprehensive understanding of the research area. The validation results indicate that users find RTMN notations simple to understand and intuitive to use.

This paper covers the second generation of the RTMN. We call this new version RTMN 2.0. RTMN 2.0 adds many additional notations: four human–robot collaboration (HRC) tasks, requirements, and KPIs; condition checks and decision making; join/split; and data association. HRC is the main focus among these notations.

This research is conducted under the research project ACROBA. ACROBA is an EU H2020 project. The consortium consists of 17 partners from 9 countries. "The ACROBA project aims to develop and demonstrate a novel concept of cognitive robotic platforms based on a modular approach able to be smoothly adapted to virtually any industrial scenario applying agile manufacturing principles" [1].

1.1. What Is Human–Robot Collaboration (HRC)?

The ever-growing advancements in the manufacturing industry have made it necessary for companies to optimize their systems by decreasing human workload, fatigue risk, and overall costs. This necessity has led to the introduction of human–robot collaboration in industrial environments [4]. Industry experts have established that the complete removal of humans from manufacturing systems is not viable. A more realistic goal is to have humans and intelligent machines working in harmony [5]. From an anthropological standpoint, "collaboration" refers to multiple parties communicating with each other and coordinating their actions in order to achieve a common goal. To this aim, collaborators observe each other, infer the intent behind an action, and plan their own actions in accordance with this intent. Similarly, in human–robot collaboration, robot systems should be designed with the appropriate tools to coordinate their actions with humans and employ the relevant cognitive and communicative mechanisms such that they can plan actions toward an established goal [6].

For the successful implementation of human–robot collaboration, the machines would require advanced cognitive capabilities to allow the human operators to collaborate comfortably and efficiently and maintain high confidence in these systems. If implemented correctly, industries may achieve a reasonable task reduction for human operators. To this end, robots should be equipped with understanding capabilities that allow them to operate with a human, just as two humans would when working together [7]. HRC systems do not rely on an equal divide of workload between humans and robots. The levels of robot automation are based on the application and are decided such that they lead to an overall improvement in the system's performance [8]. This improvement in performance can be attributed to the complementary strengths of either party. Robots offer an efficient and guaranteed performance at high speeds, whilst humans offer understanding, reasoning, and problem solving [4].

1.2. The Importance of Human–Robot Collaboration (HRC)

The aim of industrial robotics is to enable efficient performance repeatedly and accurately [9]. Currently, assembly lines have important requirements related to adaptability, mainly due to the rapid rate at which new products are introduced to the market, as well as changing technologies. The current trends in industry reflect a shift from "mass-production" to "mass-customization". Products now come with numerous variants or upgrades and a much shorter product lifetime. This imposes a challenge regarding flexibility and adaptability on the manufacturing process, a challenge for which human–robot collaboration is an attractive arrangement [10]. Companies have traditionally relied on robots with in-built capabilities and limited flexibility. However, this level of flexibility is not enough to match the current market's demands [11]. While traditional robot systems provide high payload capabilities and repeatability, they then suffer from limited flexibility and dexterity [12]. Thus, human–robot collaboration is a suitable arrangement to leverage the unique capabilities of both humans and robots for increased efficiency and quality in industrial scenarios. The recent trend of automation and data exchange, known as Industry 4.0, also supports the use of collaborative systems in industry. The aim of Industry 4.0 is to achieve efficiency, cost reduction, and increased productivity by means of integrated automation. This aim highlights the need for flexible and interoperable systems, including intelligent decision-making software and robots that can be quickly, safely, and intuitively operated by humans [13].

Industries are increasingly relying on HRC arrangements, both from an engineering perspective and from a socio-economic standpoint. While the manufacturing industry is a significant source of employment, it has been reported that most jobs offered by this sector may remain unfilled. This is attributed to a shortage of workers with the relevant technological and technical skills [14]. HRC is, therefore, a promising alternative, which makes up for the skill gap, still requires human operators, and may be more attractive to the younger generation. Additionally, robotic systems result in higher competitiveness with countries with cheap labor systems and increase trust in the company's technological aptitude. Collaborative systems also alleviate the ergonomic burden on human workers, resulting in an improved work environment and a reduction in occupational injuries. This makes environments that include both robots and human laborers more attractive to interested partners, customers, and the public [15]. Modern technologies of intuitive systems such as augmented reality, walkthrough programming, and programming by demonstration are all simple methods to operate collaborative robots, unlike the advanced technical expertise necessary to operate traditional robotic systems [13].

Currently, collaborative robotic solutions are attractive even to small- and medium-sized companies since such systems are more affordable, compact, and easy to use compared with traditional robotic systems. Traditionally, factory floors have had strict divisions of labor, with robots confined to strict safety cages far from humans. Collaborative robots overcome this division of labor, allowing humans and robots to work closely together. In doing so, the advantages of strength and automation of the robot are combined with the flexibility and intuitive nature of the human [12,13]. Evidently, there are numerous advantages to collaborative robotic systems, including economic, social, and ergonomic improvements to traditional systems. However, to harness the full benefits of such systems, companies should adhere to the appropriate safety standards to ensure optimal operation. These will be discussed in the following section.

1.3. Safety Standards and HRC Modes

The International Federation of Robotics has reported an all-time high of 517,385 new industrial robots installed in 2021 in factories around the world, with a growth rate of 31%, increased by 22% compared to 2018 (pre-pandemic record). Until now, the stock of operational robots around the globe has reached 3.5 million units [16]. With the increasing use of robots in industry, standardization and guidelines to ensure the safety of human operators are required [10]. Many standards have been proposed to give guidelines for the

safe use of collaborative robots. The machinery safety is regulated under the Machinery Directive, which covers the scope of collaborative applications [10]. The following reference standards are reported (see Table 1):

Table 1. Safety standards.

Type	Description	Standard
Type A Standard	Basic safety standards for general requirements	ISO 12100: 2010 "Machine safety, general design principles, risk assessment, and risk reduction" IEC 61508: terminology and methodology [17]
Type B Standard	Generic safety standards	– B1 standards (ISO 13849-1, IEC 62061): specific safety aspects [18] – B2 standards (ISO 13850, ISO 13851): safeguard [19,20]
Type C Standard	Safety countermeasures for specific machineryPrioritized over Type A and Type B standards	– ISO 10218: safety of industrial robots – ISO 10218-1:2011 "Robots and equipment for robots, Safety requirements for industrial robots, Part 1: Robots", safety requirements for robot manufacturers (robot and controller) [21] – ISO 10218-2: 2011 "Robots and equipment for robots, Safety requirements for industrial robots, Part 2: Systems and integration of robots", safety requirements for system integrators (robot and ancillary devices), [22] – ISO TS 15066: 2016 "Robots and robotic devices, Collaborative Robots", guidance on collaborative robot Operations [23,24]

Four categories of safety requirements are defined for collaborative robots in the type C international standards (ISO 10218-1, ISO 10218-2, and ISO TS 15066) [10,12,13]:

- Safety-rated monitored stop (SMS)

SMS [10,13] is a collaboration arrangement in which robot motion is stopped before a human operator enters the collaborative workspace to interact and carry out a task with the robotic system. This is the most basic form of collaboration and takes place within a collaborative area, that is, an area of operation shared by the robot and the human. Both parties can work in this area, but not simultaneously, since the robot cannot move if the operator is in the shared space. Therefore, it is ideal for tasks in which the robot primarily works alone and is occasionally interrupted by a human operator. Examples of such tasks include visual inspection or the positioning of heavy components by the robot for the human.

- Hand-guiding (HG)

Another mode of collaboration is known as hand-guiding [10,13], or "direct teach". In this mode, the operator simply moves the robot to teach it significant positions, without the use of intermediate interfaces such as teach pendants. These positions are communicated as commands to the robot system. Throughout this process, the robot arm's weight is compensated such that its position is held. A guiding, hand-operated device is used by the operator to guide the robot's motion. For this advanced form of collaboration, the robot must be equipped with safety-rated monitored stop and speed functionalities. Once the robot has learned the motion and the human operator has left the collaborative area, the robot may execute the program in automatic mode. However, if the operator enters the area, the program is interrupted. When the operator is using the hand-guiding device, the robot operates in a state of safety-rated monitored speed functionality until the operator releases the arm and leaves the collaborative area, allowing the robot to resume automatic operation once again.

- Speed and separation monitoring (SSM)

In SSM [10,13], the robot operates even when a human is present by means of safety-rated monitoring sensors. Thus, human and robot operations take place simultaneously. To reduce risks, a stipulated protective distance must always be kept between the two parties. If this distance is not kept, the robot operation stops and only resumes once the operator

has moved away from the system. If the robot system operates at a reduced speed, the protective distance is reduced accordingly. The workspace may be divided into "zones", whereby if the human is in the green zone, the robot may operate at full speed, if in the yellow zone, the robot operates with reduced speed, and if the human enters the red zone, the robot's operation is stopped. Vision systems are used to monitor these zones.

- Power and force limiting (PFL)

PFL [10,13] is a collaborative approach in which limits are set for motor power and force such that the human operator and robot may work side-by-side. These limits are set as a risk reduction method, defined by a risk assessment. To implement this approach, specific equipment and control modes are required in order to handle collisions between the robot and human and prevent any injuries to the human.

These four collaborative modes can be applied to both traditional industrial robots and collaborative robots. For traditional industrial robots, additional safety devices such as laser sensors or light curtains are required. On the other hand, for collaborative additional features such as force and torque sensors, force limits, vision systems, laser systems, and anti-collision systems are required [12].

1.4. HRC Task Types

It is important to analyze the different types of collaboration tasks [10,12,13]. Matheson et al. used the classification that Müller et al. [25] proposed for human–robot collaboration in their paper [12], which distinguishes HRC task types into four groups: coexistence (same environment, no interaction), synchronized (same workspace, different times), cooperation (same workspace, at the same time, separate tasks), and collaboration (same workspace, same task, same time). Wang et al. [26] presented the following types in their paper: coexistence (not sharing workspace, no direct contact), interaction (sharing workspace, communicating with each other, performing tasks sequentially), cooperation (sharing workspace, having individual goals, sharing resources, working simultaneously), and collaboration (joint activity, sharing workspace, having the same goal, physical contact allowed). Thiemermann [27] differentiated four operating modes: manual mode (human), automation (robot), parallelization (same product, direct contact, suitable for pre-assembly), and collaboration (same product, work together). There are other classifications in the literature [28–30]. To summarize, there are four basic task types for HRC based on the literature: coexistence, sequential cooperation, parallel cooperation, and collaboration.

2. Materials and Methods

2.1. Literature Review and Analysis on HRC Modeling Methods

Choosing the appropriate modeling language is imperative for the development of a robust human–robot collaboration system. This choice depends on various factors, including the particular task requirements, the target platform or framework, and the specific preferences and technical skills of the developers. Commonly used modeling languages in this field include Business Process Modeling Notation (BPMN), Systems Modeling Language (SysML), behavior trees (BT), Unified Modeling Language (UML), and Petri Nets. Literature related to each of these languages is reviewed and discussed in this section.

2.1.1. Business Process Modeling Notation (BPMN)

BPMN was originally developed as a standard notation for business users to bridge the gap between business process design and process implementation. A Business Process Model contains a network of graphical objects representing activities or work and the flow controls representing the order of performance [31]. While BPMN is typically used for business processes, its capabilities can be extended to the field of robotics. More specifically, it can be used to model the flow, tasks, decisions, and interactions that occur in HRC processes.

One approach found that there was a lack of consideration for the modeling of collaborative tasks and therefore attempted to completely model processes such as assembly workflows, interactions, and decisions made by machines and humans. It supports variability modeling, making it attractive for customized manufacturing scenarios. A BPMN-based workflow designer was implemented to allow the user to model processes intuitively [32].

BPMN was also used for the development of a risk analysis software tool to support changes in adaptive collaborative robotic scenarios [33]. The work presents a tool that can automatically identify any changes that have been made to a particular application's components or processes, providing safety experts with a tool to monitor manufacturing processes. The results showed that this technology provided better usability and decreased errors when compared with conventional methods. BPMN was used as a basis for development, with certain modifications to make it appropriate for collaborative scenarios.

Another paper tackles the task of warehouse material handling using an automated guide vehicle system [34]. The Manufacturing Process Management System (MPMS) is used to control the process. The aim is to achieve a system that performs all tasks automatically, without requiring major changes for various processes. The MPMS was designed using BPMN and Camunda as a platform, allowing for the BPMN model to be executed and controlled in real time.

Concerning the lack of efficiency arising from static task allocation, one paper focused on the use of adaptive task sharing in manufacturing and assembly [35]. A model was developed for experimental purposes, which included task sharing worker assistance software based on BPMN. The results indicated that adaptive task sharing reduced the productivity gap in automated assembly.

2.1.2. Unified Modeling Language (UML)

UML is a popular modeling language in the fields of software engineering and system design. Through its standardized notation, clear visualization and specification of structures, behaviors, and interactions of systems is achieved. UML is, therefore, thought to be an ideal approach to model architecture, interfaces, and interactions in HRC systems. The use of UML in the field of human–robot collaboration seems to mostly focus on safety and risk analysis. This can be seen in [36–39].

F. D. Von Borstel et al. [40] used a combination of UML and colored Petri Nets (CPNs) to model and simulate mobile robots based on wireless robotic components (WRCs). UML diagrams were customized to describe the specific WRC architecture based on the robotic software, particular task, and operational environment. The hierarchical CPNS were developed using the customized UML diagrams as a guide. From there, an executable model of the robotic system could be developed. Further work will aim to improve the translation from UML to CPNs.

L. Carroll et al. [41] attempted to use UML in development for the design of real-time robot controllers, based on the necessity of better adaptability from robots when working in cooperation with humans. The main focuses were dependability, fault avoidance, fault removal, and fault tolerance. UML was found to be appropriate for the development and maintainability of this controller.

As a major software engineering modeling tool, UML can model requirements and high-level system architecture very efficiently. However, UML is not well-established for robotics, including HRC. While UML can be useful for modeling certain aspects of human–robot collaboration, limitations arise when applied to the full scope collaborative scenarios. Robots often operate autonomously and need to make decisions, but UML does not offer specialized modeling constructs for representing autonomy or decision-making processes.

2.1.3. Systems Modeling Language (SysML)

SysML is a standardized formal language, of which the main purpose is the modeling of systems engineering processes. It is based on UML but is more well-adapted for complex systems in engineering and automation.

Researchers proposed SysML as a modeling language as it was thought to simplify robot programming through abstraction [42]. Since robots are increasingly being integrated into environments where humans are present, they should be made safer, with straightforward programming and interaction capabilities. To achieve this, SysML was used to represent manufacturing processes by graphical diagrams capturing the system's structure from various perspectives. While this work did not focus directly on modeling human–robot collaboration, it was one of the motives for the approach, and the capabilities of the system show promise to be adapted for HRC scenarios.

While the literature specifically focusing on using SysML for HRC tasks is limited, SysML has been widely used for modeling general robotic workflows. K. Ohara et al. [43] proposed robot software design based on SysML. The language was chosen due to its benefits in terms of reusability and flexibility. R. Candell et al. [44] aimed at achieving real-time observation and control using SysMl modeling for architecture, components, and information flows. This was based on Industry 4.0 and the resultant requirements for adaptability, flexibility, and responsive communication. SysML is used to develop a graphical model for a wireless factory work cell.

In summary, although SysML can be a useful tool for modeling certain aspects of HRC such as requirements, system architecture, and interaction with external systems, SysML is not the most suitable tool for modeling real-time interaction or human and complex interaction scenarios.

2.1.4. Behavior Trees

Behavior trees have been widely used in the field of robotics to define the behavior of autonomous agents. The graphical modeling language involves a hierarchical structure of tasks and dependencies, making them an ideal approach to model complex decisions and coordination in HRC scenarios.

The literature related to behavior trees is largely concerned with the limitations of automation to capture human dexterity and judgement, especially in unstructured environments where unforeseen events commonly occur. A framework was thus presented, which captured various behaviors through which robot autonomy is enhanced. This was achieved using behavior trees to model the robot's intelligence, such as social behaviors, human intention, and various tasks, including collaborative ones [45].

While behavior trees can be a valuable tool for modeling the autonomous behavior of a robotic system, they are less suited for modeling the complexities of human–robot collaboration that involve, for example, communication, shared decision-making, and social interaction. Behavior trees tend to focus on predefined actions and decision sequences, making them less flexible in terms of accommodating the varied and sometimes unpredictable nature of human intention. Behavior trees prioritize system-centric decision-making and are not adept at handling human-centric decision processes. The design of behavior trees is not human-centered, nor user-friendly; this is another drawback of behavior tree modeling.

2.1.5. Petri Nets

Petri nets were developed primarily for the modeling of systems in which events may occur concurrently, with constraints on the concurrence [46]. Petri nets for human–robot collaboration were found in the literature. A. Casalino et al. [47] presented a scheduling method for collaborative tasks in assembly, which aims at achieving optimal planning and adaptability of assembly process based on runtime knowledge. Similar work was carried out in refs. [48,49]. Colored petri nets were used to model the concurrent scenario in which a human works collaboratively with a wearable robot. The developed model assigned tasks to the robot that were considered too laborious for the human in a panel installation task [50]. The main aim of the work carried out by R. E. Yagoda et al. [51] was to develop a technique to model human–robot interaction (HRI). A. Casalino et al. [52] demonstrated this in a realistic scenario involving a human operator and dual-arm robot performing an assembly task using time petri nets. However, the control policy is case-specific and needs

to be extended to adapt to general cases in future research. Petri nets were found to be advantageous, as they are generally simple yet are built upon underlying mathematics, making them powerful for such modeling applications. Additionally, their graphical nature makes them a great communication medium.

Petri nets have limitations on expressiveness for complex decision making, task allocation, and communication. Their design is mainly used for discrete event modeling other than real-time and continuous motion and perception. Petri nets also do not naturally integrate with AI and learning algorithms. They are not considered the most suitable choice for modeling the complexity and dynamics of human–robot collaboration, which involves a combination of continuous and discrete interactions, uncertainty, and complex decision-making processes.

2.1.6. Research Gap

To summarize, UML and SysML are more well-adapted for high-level system modeling, requirement, and system integration. They are more geared for technical people instead of systems' end users. Behavior trees then mainly focus on automation, predefined actions, and decision sequences, which are not suitable for humans nor HRC. Petri nets lack expressiveness for decision making and communication. Although BPMN notations have been found to be well-accepted by its intuitive modeling notation in terms of usability, the standard BPMN focuses on business process modeling and does not have separate notations for robots (robot tasks, robot skills, robot primitives, and HRC tasks). Although BPMN can be applied to robotics, the process model becomes too complicated for non-experts in BPMN. In general, there is a need to model HRC, but these aforementioned modeling tools are not suitable for this purpose. They are all well-established tools in their application areas. However, they are not developed specifically for robotics in the HRC domain. In the current versions of these tools, it is not possible to separate the HRC task types, specify the HRC modes, verify safety standards, nor monitor human factors.

Therefore, we have created a new modeling language, RTMN, for robotic processes that is based on BPMN. It uses the well-known basic notation shapes and builds robotic notations upon them. After our first RTMN version, we continued our research with the aim of improving RTMN, especially in the human–robot collaboration area. RTMN 2.0 is an extension of RTMN.

2.2. RTMN 2.0—Extension of RTMN

In this section, we will explain the RTMN 2.0 in detail. First, we will introduce the former RTMN elements and then the additional elements in RTMN 2.0. Afterwards, we will present the RTMN 2.0 sequence flow connection rules. The HRC model follows this and is the core contribution of this paper. Later, other extensions and modifications of RTMN are added. Finally, the implementation and demonstration of RTMN 2.0 are described at the end of this chapter.

2.2.1. The RTMN Elements

RTMN originally contained eleven basic notations differentiated by shapes, colors, and icons [2]. These notations were developed based on the well-known business process modeling notation (BPMN) [53]. The goal was to develop a modeling language that covers the robotics domain with familiar notations, rather than completely new ones. These notations adopt colors, icons, and shapes in their design, aiming to make the modeling more intuitive. Four BPMN standard notations, "sequence flow", "event" (start, end), and "gate way" (exclusive), are reused in RTMN (see BPMN specification [53]).

There are five new robotics-specific notations: robot task, robot skill, robot primitive, HRC (human–robot collaboration) task, and one slightly changed notation—human task. It is similar to a user task in BPMN; however, here, it has a human icon and a color assigned to it. A "task" in BPMN is the lowest level of the process and cannot be broken down into finer details. It is considered an atomic activity within a process [3]. However, the

term "task" in the robotics area is, rather, a subprocess in relation to BPMN. With different usages—a picking task, for example—it is not an atomic activity but consists of detailed steps such as to move to the right place, locate the gripper and grasp, etc. Therefore, we differentiate a robot task from a typical BPMN task. We define it as an activity consisting of a set of activities that corresponds to a special type of BPMN sub-process that deals with robot activities. The activities within the robot task are defined as skills and primitives. Primitives are atomic activities that compose robotic skills, and skills form robot tasks.

2.2.2. The RTMN 2.0 Elements

RTMN 2.0, in comparison to the first version of RTMN, has the following extensions:

- Adding HRC modeling elements, including safety in combination with collaboration modes and task types of humans and robots. This has significantly enriched the former RTMN model and enabled it to be applied to HRC application areas.
- Adding requirements, with KPI as a basic element.
- Adding the link from requirements/KPI to robot control.
- Adding decision-making elements.

RTMN 2.0 was designed especially for modeling human–robot collaboration processes. Together with version 1.0, it is capable of modeling light-out automated robotic processes as well as human–robot collaboration processes. The model contains two sets of elements: the RTMN elements and the extended elements. A complete overview is presented in Figures 1–3. It consists of all the elements of RTMN 2.0, with the newest improvements and extensions.

Notation	Notation Name	Description	Attribute	Condition	Cardinality	Relation
	Process	Process is a complex activity or set of activities that accomplish a specific organizational goal. It is a orange (#ED7D31) double line rectangle with round corners	has name has KPI has requirement has goal has task has sequenceElement has inputs (goals) has outputs (results) has feedback has precondition has postcondition	N/A	has at least one task	a process can be connected to another process
	Human task	It is an activity that is performed by a human It is a green (#009688) rounded corner rectangle with a single thin line and includes a human figure marker in the top middle of the rectangle	has name has human performer has inputs has output (button/signal)	no robot involvement	has at least 1 human performer	output is used to continue the process
	Robot Task	Robot task is a compound activity that done by robots to perform a complex robot action. It can be broken down into finer level of details. It is a red (#FF5252) rounded corner rectangle with a single thin line and includes a robot figure marker in the top middle of the rectangle, a "plus" sign in the lower-center of the shape indicating that the activity has a lower level of details.	has name has sequenceElement has inputs (goals) has outputs (results) has feedback has precondition has postcondition	no human involvement	no human operator involved.	has skills &/robot task. Used to compose other robot task and processes
	Skill	Robot skills are used to integrate and syncHRConize robot actions and sensor data in a consistent way. A robot skill is an elementary functionality or action provided by a robot in order to perform an autonomous mission. It is a blue (#448AFF) rounded corner rectangle with a single thin line and includes a robot figure marker in the top middle of the rectangle, a "plus" sign in the lower-center of the shape indicating that the activity has a lower level of details.	has name has skill has primitive has sequenceElement has inputs has outputs has feedback has precondition has postcondition	no human involvement	has at least 1 primitive	has primitives &/ skills Used to compose robot tasks
	Primitive	Primitives are simple, atomic robot activities that can be combined to form more complex robot behaviors. They are the lowest level model elements and provide direct access to hardware elements. It is a green (#4CAF50) rounded corner rectangle with a single thin line and includes a robot figure marker in the top middle of the rectangle	has name has inputs has outputs has feedback has precondition has postcondition	no human involvement	atomic, cannot be broken down to a finer level of detail	receives information from other primitives and skills. Used to compose robot skills

Figure 1. RTMN modeling elements—Part A: basic notation.

2.2.3. The RTMN 2.0 Sequence Flow Connection Rules

The authors display the rules regarding how to connect the elements in a sequence flow in Figure 4. The basic rules are as follows: Start can only start a process and cannot have any element flow into it. End is the opposite; it ends the process and cannot have any elements flow out of it. Sequence Flow connects the other elements and points out the direction of the process flow. Requirements are normally defined on the process level, but they can also be assigned to tasks, skills, and primitives if needed. KPIs are usually used to measure the achievement of requirements for the process but can also be assigned to tasks, skills, and primitives if needed. Tasks can be connected to any other task, condition,

decision, or sequence flow. Skills and primitives can be connected to skills and primitives as well as conditions, decisions, and sequence flows.

	Task	Description	Attributes			
	HRC Task - Coexistance Fence (CF)	Human robot task CF is a compound activity that is performed by both human and robot. It can be broken down into finer level of details. This HRC task type is a combination of coexistance task type and a minimum HRC mode Fence. It is a round corner rectangle with a single thin line. It has two colours the red (#FF5252) represents robot and the green (#009688) represents human. It has a white space in between indicates the robot and human do not work at the same workplace. The thick black line indicates robot in fence. It has a "plus" sign in the lower-center of the shape indicating the activity has a lowerlevel of detail.	has name has sequenceElement has inputs (goals) has outputs (results) has feedback has precondition has postcondition has loopCharacteristics has HRCTaskType has HRCMode has Safety Level			
	HRC Task - Sequential Cooperation SMS(SS)	Human robot task SS is a compound activity that is performed by both human and robot to achieve a common goal. It can be broken down into finer level of details. This HRC task type is a combination of sequential cooperation task type and HRC mode SMS. It is a round corner rectangle with a single thin line. It has two colours the red (#FF5252) represents robot and the green (#009688) represents human. It has an icon indicating the sequential cooperation task type and HRC mode SMS. It has a "plus" sign in the lower-center of the shape indicating the activity has a lowerlevel of detail.	has name has sequenceElement has inputs (goals) has outputs (results) has feedback has precondition has postcondition has loopCharacteristics has HRCTaskType has HRCMode has Safety Level	human and robot are both involved in completing the activity	has at least 1 human task and 1 robot task	has human task and robot task has HRC task type: sequential has HRC mode: SMS has Safety level: >=1
	HRC Task - Teaching HG(TH)	Human robot task TH is a compound activity that performed by both human and robot to achieve a common goal. It can be broken down into finer level of details. This HRC task type is a combination of teaching task type and HRC mode HG. The middle brown area with the joined color of red and green represents the shared workspace by human and robot. It is a round corner rectangle with a single thin line. It has two colours the red (#FF5252) represents robot and the green (#009688) represents human. It has an icon indicating the collaboration task type and HRC mode. It has a "plus" sign in the lower-center of the shape indicating that the activity has a lowerlevel of detail.	has name has sequenceElement has inputs (goals) has outputs (results) has feedback has precondition has postcondition has loopCharacteristics has HRCTaskType has HRCMode has Safety Level	human and robot are both involved in completing the activity	has at least 1 human task and 1 robot task	has human task and robot task has HRC task type: teaching has HRC mode: HG has Safety level: >=2
	HRC Task - Parallel Cooperation SSM(PS)	Human robot task PS is a compound activity that performed by both human and robot to achieve a common goal. It can be broken down into finer level of details. This HRC task type is a combination of parallel cooperation task type and HRC mode SSM. The middle brown area with the joined color of red and green represents the shared workspace by human and robot. It is a round corner rectangle with a single thin line. It has two colours the red (#FF5252) represents robot and the green (#009688) represents human. It has an icon indicating the collaboration task type and HRC mode. It has a "plus" sign in the lower-center of the shape indicating that the activity has a lowerlevel of detail.	has name has sequenceElement has inputs (goals) has outputs (results) has feedback has precondition has postcondition has loopCharacteristics has HRCTaskType has HRCMode has Safety Level	human and robot are both involved in completing the activity	has at least 1 human task and 1 robot task	has human task and robot task has HRC task type: parallel has HRC mode: SSM has Safety level: >=3
	HRC Task - Collaboration PFL(CP)	Human robot task CP is a compound activity that performed by both human and robot to achieve a common goal. It can be broken down into finer level of details. This HRC task type is a combination of collaboration task type and HRC mode PFL. It is a round corner rectangle with a single thin line. It has two colours the red (#FF5252) represents robot and the green (#009688) represents human. It has an icon indicating the sequential collaboration task type and HRC mode SMS. It has a "plus" sign in the lower-center of the shape indicating that the activity has a lowerlevel of detail.	has name has sequenceElement has inputs (goals) has outputs (results) has feedback has precondition has postcondition has loopCharacteristics has HRCTaskType has HRCMode has Safety Level	human and robot are both involved in completing the activity	has at least 1 human task and 1 robot task	has human task and robot task has HRC task type: joint has HRC mode: PFL has Safety level: >=4

Figure 2. RTMN modeling elements—Part B: HRC tasks.

2.2.4. The HRC Model

The authors have developed an HRC model that covers the following aspects of HRC: HRC task types, HRC modes, combining the HRC task types and HRC modes, workspace, and decision making. The authors proposed hardware and software solutions for each HRC task type. The authors address every aspect one by one in the following sections.

Combining Collaboration Task Types and HRC Modes

Based on the literature ([10–16,25–30]), the authors consolidated the different classifications of the HRC task types and HRC modes. In the literature, five HRC modes (fence, SMS, HG, SSM, and PFL) and five HRC task types (cell, coexistence, synchronized, cooperation, and collaboration) are commonly defined. The HRC modes can ensure the safety of people even when unexpected human intrusions occur. However, the HRC task types do not by definition ensure the safety of people. This classification focuses on how closely humans and robots work together, and only the people planned for the task are considered. In reality, no matter how the authors define the task type, humans can act differently than expected, and human intrusions can occur. Therefore, to ensure safety, each task type was required to have a minimum safety mode enabled. For this reason, the authors present a

new classification that integrates the HRC task types, HRC modes, and safety levels. Safety level 0 indicates no risk to the human from the robot. It has the lowest safety requirements, as the robot is within a physical fence. The safety levels go up to level 4, which requires the highest safety requirements not only from hardware but also from software. The authors also distinguish the collaboration levels (0–5): the higher, the more collaboration between human and robot.

		Description	Attributes	Rule/Decision table	Constraints	Connection
	Requirement	Requirements are KPI targets to achieve certain process goals.	has name has requirementDes has goal has KPI has formula has input has output has precondition has postcondition	N/A	at least one	connecting to process or subprocess
	KPI	KPI stands for key performance indicator, a quantifiable measure of performance over time for a specific objective.	has name has KPIDes has formular	N/A	quantitative KPI, at least one	connecting to requirement
	Sequence Flow	connect flow objects in a process, the arrow direction indicates the workflow direction	has name has sourceRef has targetRef	N/A	can connect only one element on the left and one on the right	connecting process elements
	Data Association	connect data input or output to a flow object in a process.	has name has sourceRef has targetRef	N/A	can connect only one element on the left and one on the right	connecting process elements
	Condition	Condition is rule-based decision making. It is enabled by a rule table where users can specify their decision logic. It is a dimond shape with a rule table icon.	has name has ruleTable has alternativePath	Rule table	system can execute automatically	is connected with multiple sequence flows
	Decision	Decision refers to human decision making. Human decision making allows operators/users to specify a question that will be answered at run time. It is a dimond shape with a human icon.	has name has condition has alternativePath has humanInput	Decision table	Human needs to decide here by giving input	is connected with multiple sequence flows
	Workspace	Workspace is an area in the robotic cell/workplace. It is used to seperate the different physical spaces. It is a black rectangle with a smaller rectangle in it on the left.	has name has inputs has outputs has feedback has precondition has postcondition			
	Start	indicating where a process start	has name	N/A	can only be used to start a process	can be connected to a sequence flow
	End	indicating where a process will end.	has name	N/A	can only be used to end a process	a sequence flow can be connected to it

Figure 3. RTMN modeling elements—Part C: other notations.

Figure 4. RTMN 2.0 sequence flow connection rules.

For each task, the authors have defined general safety requirements. A solution design for the hardware and software level is proposed for each HRC task type. This makes it easier for process engineers to ensure safety when implementing HRC processes. The overall approach is presented in Figure 5.

Figure 5. Combining HRC task types and HRC modes [21,22].

In our new classification, we have five categories: coexistence fence (CF), sequential cooperation SMS (SS), teaching HG (TH), parallel cooperation SSM (PS), and collaboration PFL (CP). The "cell" of the human robot task type refers to the robot working autonomously, surrounded by physical fences without human involvement, so we only needed to model the robot tasks and no human tasks need to be modeled. In this case, we do not need to create a separate HR task type for it, as it has only robot tasks. We added a new task type, "Teaching", which is when a human takes control of the robot and demonstrates what the robot should do, then the robot learns and repeats it. The reason for adding it is to cover this missing collaboration task type and then to be able to map it to the collaboration mode HG.

- Coexistence Fence (CF)

This type (Figure 6) combines the task type "coexistence" and the collaboration mode "fence". This task has the lowest HRC level 0 and should activate at least safety level 0 (fence) in the background. We recommend using an industrial robot, since humans and robots do not work together in the same workplace. The fence for the robot can be in the form of a physical barrier or certified hardware such as a light curtain/2D scanner. As long as fence mode is insured, this task is safe. No additional software functionalities are needed.

- Sequential Cooperation SMS (SS)

This type (Figure 7) combines the task type "synchronized" and collaboration mode "safety-rated monitored stop". "safety-rated monitored stop" means "condition where the robot is stopped with drive power active, while a monitoring system with a specified sufficient safety performance ensures that the robot does not move", as described in [54].

This task has collaboration level 1 and should activate at least safety level 1 (SMS) in the background. We recommend using either an industrial robot or a collaborative robot with safety-certified hardware, such as a light curtain/2D scanner to ensure SMS (robot

stops when human enters the robot working area). As long as SMS mode is ensured, this task is safe. No additional software functionalities are needed.
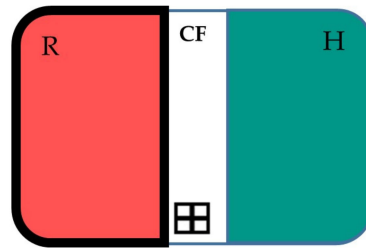

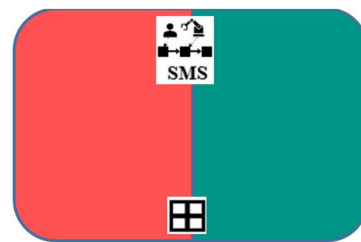**Figure 6.** Coexistence fence notation.

Figure 7. Sequential cooperation SMS notation.

- Teaching HG (TH)

This type (Figure 8) combines the task type "teaching" and collaboration mode "Hand Guiding". This task has collaboration level 2 and should activate at least safety level 2 (HG) in the background. We recommend using a collaborative robot with safety-certified hardware that allows the robot's state to switch to safety-rated monitored speed functionality to allow for the direct movement of the robot, e.g., a hand guiding device. If a minimum safety HG mode is ensured, this task is safe. Safety-rated monitored speed functionalities need to be programmed at the software level. In ISO 10218-1, safety-rated monitored speed is written as "safety-rated function that causes a protective stop when either the Cartesian speed of a point relative to the robot flange (e.g., the TCP), or the speed of one or more axes exceeds a specified limit value" [54].

- Parallel Cooperation SSM (PS)

This type (Figure 9) combines the task type "parallel cooperation" and the collaboration mode "speed and separation monitoring". This task has collaboration level 3; it should activate at least safety level 3 (SSM) in the background and put in place the hardware that ensures the SSM mode. We recommend using a collaborative robot with safety-certified hardware that has zone/position control capabilities, e.g., a 2D scanner, pressure pads/floors, or a 3D safety vision scanner. Speed/position control functionalities need to be programmed at the software level. The work area should be divided into three zones: green zone (robot runs at full speed), yellow zone (robot runs at reduced speed), and red zone (robot stops). As long as SSM mode is ensured, this task is safe.
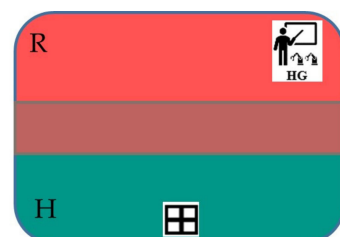

Figure 8. Teaching HG notation.

- Collaboration PFL (CP)

This type (Figure 10) combines the task type "collaboration" and the collaboration mode "power and force limiting". This task has collaboration level 4 and should activate safety level 4 (power and force limiting) in the background. This type is the most collaborative task type, requires the highest safety controls, and demands the most software engineering effort. We recommend using a collaborative robot with safety-certified hardware that can control the limitations of motor power and force, e.g., a 3D safety vision scanner or a vision system. For force control and limitation, force/torque sensors in end-effectors and/or in the joints of the robot are required. At the software level, functionalities need to be developed for collision handling, distance monitoring, force control, velocity tracking, and power of robot limitation. As long as PFL mode is ensured, this task is safe. For this notation, one must assign HRC mode 4 and task type collaboration to it.

Figure 9. Parallel cooperation SSM notation.
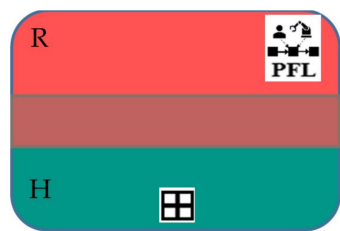
Figure 10. Collaboration PFL notation.

We calculated the minimum separation distance between the robot and the human (D_{HR}) with the following parameters: Speed Robot (V_R), Speed Human (V_H), Reaction Time (T_R), and Stop Time (T_S). An illustration of the formula is shown in Figure 11.

$$D_{HR} = V_R * (T_R + T_S) + V_H * (T_R + T_S)$$
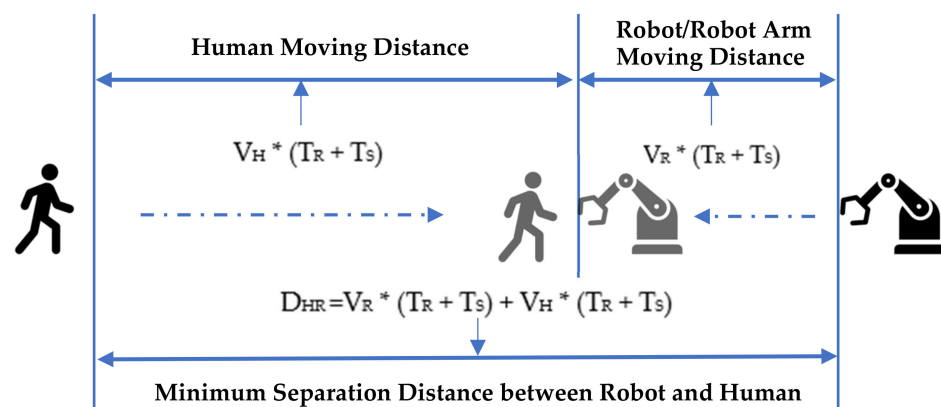
Figure 11. Minimum separation distance.

Workspace

To align the modeling with a realistic robotic cell, we introduced the workspace notation. This notation is similar to the BPMN pool/lane (see Figure 12). In BPMN, pools represent companies, departments, or roles. Lanes represent sub-entities within these

organizations and appear as swim lanes inside the pool [3]. For robotic processes, there are usually only limited humans involved in the process. In the robotic cell, the focus is not different entities, such as roles or departments, but different areas in the cell. Therefore, we propose the use of pools/lanes as workspaces for separating the different areas in the robotic cell.

Figure 12. Workspace notation.

One can use workspaces when tasks involved in the process are carried out in multiple workspaces. When tasks switch from one workspace to another, there will be a pop-up window or a voice reminder of the workspace change. A workspace can have input and output and precondition and postcondition assigned to it. With these properties, one can set up specific requirements for the workspace. An example of the use of workspace notation is shown in Figure 13.
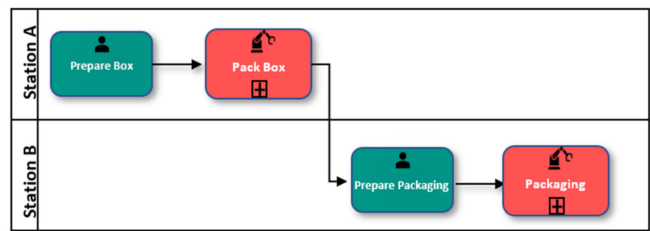
Figure 13. Example of using workspace notation.

Decision Making

We modeled decision making with two different notations. We called one "condition" (see Figure 14), which represents rule-based decision making, and the other "decision" (see Figure 15), which refers to human decision making.
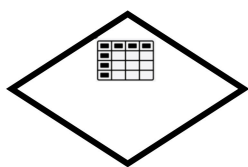
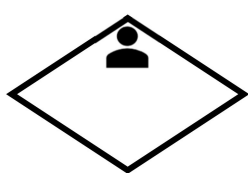**Figure 14.** Condition notation.

Figure 15. Decision notation.

Rule-based decision making is enabled by a rule table where users can specify their decision logic. We illustrate in Table 2 how the rule table looks, with some examples to demonstrate its usage.

The rule table checks whether the conditions are met. Depending on whether the conditions are met or not, a corresponding action is followed. Often, there is only one parameter that is being checked, for example, "quality check". However, there are also cases where multiple parameters are checked. In this case, we need to define the sequence of the condition check. We use a sequence column to indicate that the parameters are

checked sequentially one by one. When the rule table is defined, this sequential execution must be followed.

Table 2. Rule table.

Sequence	Parameter	Equation	Value	Action
1	Bin status	$=$	Empty	Fill up bin
		$\neq$	Empty	Do nothing
2	Distance, robot and human	$=$	$V_R * (T_R + T_S) + V_H * (T_R + T_S)$	Stop robot
		$>$	$V_R * (T_R + T_S) + V_H * (T_R + T_S)$	Robot run at collaborative speed
3	Quality check	$=$	Accepted	Move product to next station
		$=$	Rejected	Move to rejected bin

Human decision making allows operators/users to specify a question that will be answered at the run time. In this step, a question will pop up and ask the operator/user to give an input (commonly a yes or no answer). Based on what the operator chooses (yes or no), a different branch of the process flow will be executed. Figure 16 gives an example of the decision notation.
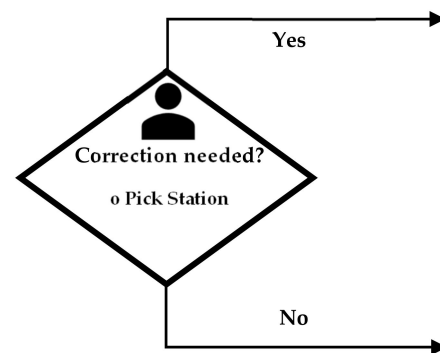
Figure 16. Example of using decision notation.

2.2.5. Other Extensions and Modifications

RTMN 2.0 not only extends the HRC area but also covers other additional functionalities. Requirements and KPIs have been added to the modeling language. The aim is to achieve traceability from requirements through tasks, skills, and primitives to reach the defined KPIs. The process notation is added to represent the different processes. We also made some modifications to the skill and primitive notations in the first version of RTMN so that the distinction would be more obvious.

Requirements and KPI

Key performance indicators (KPIs) form a specialized set of metrics steering management activities towards improving company performance, representing quantitative measures of critical success factors (CSFs) in enterprises [55]. They serve as variables reflecting progress toward goals while aligning with the company's vision and strategies [55]. Displaying KPIs is essential in order for supervisors and employees to seek improvements and reach for better performance [55]. In an HRC environment, safety and ergonomics are widely acknowledged as crucial pillars, while in a business context, productivity, encompassing both effectiveness and efficiency, and economics must be considered [55].

Therefore, after analyzing numerous research papers, Caiazzo et al. divided the various KPIs related to HRC into the following four areas: "Productivity", "Economics", "Safety", and "Ergonomics" [55]. We adopted their systematic approach and categorized the requirements and KPIs into these four categories.

The requirement notation (see Figure 16) and KPI notation (see Figure 17) were added to RTMN 2.0 to visualize the business requirements and KPIs at the process level. We could also assign requirements and KPIs at the task level if needed.
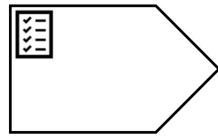
Figure 17. Requirement notation.

The requirement notation (see Figure 17) represents the business requirements of a robotic process/task. It has the following properties: name, formula, KPI, input, output, precondition, and postcondition. One can set constraints using preconditions and postconditions. For example, one can add a postcondition—the sum of net machine utilization and net operator actuation is 100%. The KPI notation defines the key performance indicators linked to the requirements (see Figure 18). Table 3 illustrates the general KPIs for the HRC robotic processes, among which a few are related to humans: accident rate, net operator actuation, and human exposure to chemicals.

Figure 18. KPI notation.

Table 3. Requirements and KPIs.

| Requirement	Requirement Formula	Name of KPI	KPI Formula
Decrease cycle time by 10%	(Cycle Time old − Cycle time new)/Cycle Time old ≤ 10%	Cycle Time	Cycle Time = Process end time − Process begin time
Increase productivity by 20%	(Productivity old − Productivity new)/Productivity old ≥ 20%	Productivity	Units of product/production time (hours)
Reach success rate of 80%	Success Rate ≥ 80%	Success Rate	Nr of success action/Nr of total actions
Reach reprogramming time of less than 10 min	Reprogramming Time < 10 min	Reprogramming Time	Time finish reconfiguration − Time start reconfiguration
Reduce scrap product to less than 5%	Nr of scrap product/Nr of total product ≤ 5%	Scrap Product	The number of products not accepted in quality inspection
Increase machinery utilization to more than 75%	Net Machine Utilization ≥ 75%	Net Machine Utilization	Machine run hours per process/process duration
Reduce net operator actuation to less than 25%	Net Operator Actuation ≤ 25%	Net Operator Actuation	Human working hours per process/process duration
Reduce accident rate to 0	Accident Rate = 0	Accident Rate	The number of reportable health and safety incidents per month
Increase human safety/reduce exposure to chemicals/danger	1 − Human exposure to chemical rate ≥ 30%	Human Exposure to Chemicals	Time of human exposure to chemicals/danger/cycle time

Traceability of Requirements and KPIs

The inputs for calculating the KPIs come from ROS, the robot control level. From primitives and skills, we can obtain the input values, use them for calculating the KPIs, and validate the requirements. Through this, traceability of the requirements and KPIs is achieved.

To be more specific, KPIs are calculated automatically based on the values created in different layers of the robot system. For example, for a pick skill, it is tracked whether the pick is successful or not. The success rate of picking is calculated by the number of success picks/total pick attempts. Another example is the robotic task scrap product, through which quality inspections of the number of accepted products and unaccepted products are tracked. If the product is within the error tolerance, it will be accepted, and vice versa. The system can add up the number of accepted products and the number of unaccepted products, and then product quality is calculated by dividing the number of products not accepted by the number of total products.

Robotic Process

A robotic process involves robots in the process. We have added this notation (see Figure 19) because we want to visually show the relationship between business requirements and the overall process. A process can contain human tasks, robot tasks, and HRC tasks.
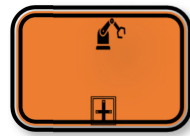
Figure 19. Robotic process notation.

The robotic process has data properties: name, owner, and organization. Requirement/Objective and KPIs are no longer just properties of the process. They have their own notations, as described earlier.

We provide an example of modeling KPIs and requirements for a process in Figure 20. The dashed arrows indicate data association.
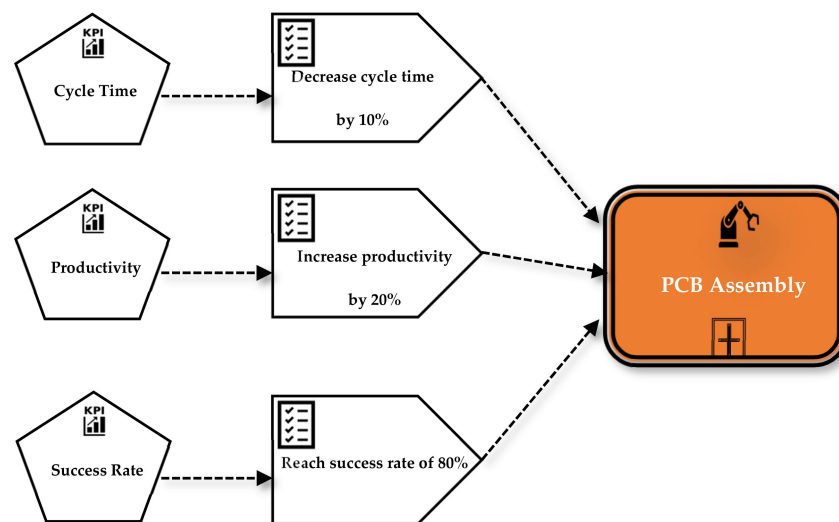
Figure 20. Process–requirement–KPI example.

Skill and Primitive

We modified the icons of the skill and primitive notation in order to easily distinguish the robot task and robot skill visually whilst bringing the appearance of the skill and primitive notation closer together. Then, both skill and primitive had robot body icons (see Figures 21 and 22). The robot task has a robot arm icon.
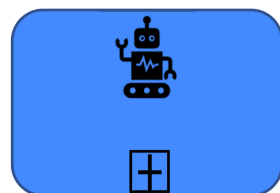
Figure 21. Skill notation.

2.2.6. The Implementation

As described in the former RTMN paper [2], the modeling language RTMN was used in the graphical user interface (GUI) of the ACROBA platform. The notations from RTMN 2.0 were defined as drag and drop elements in the GUI. The GUI was developed according to the Flutter framework [56]. Flutter uses Dart as its programming language. The special

package Dartros [57] enables Flutter to communicate with ROS [58] by implementing the ROS client library. The overall architecture is shown in Figure 23 (taken from the former paper).
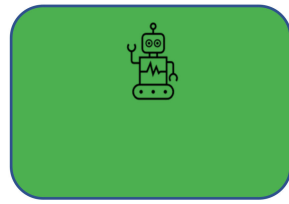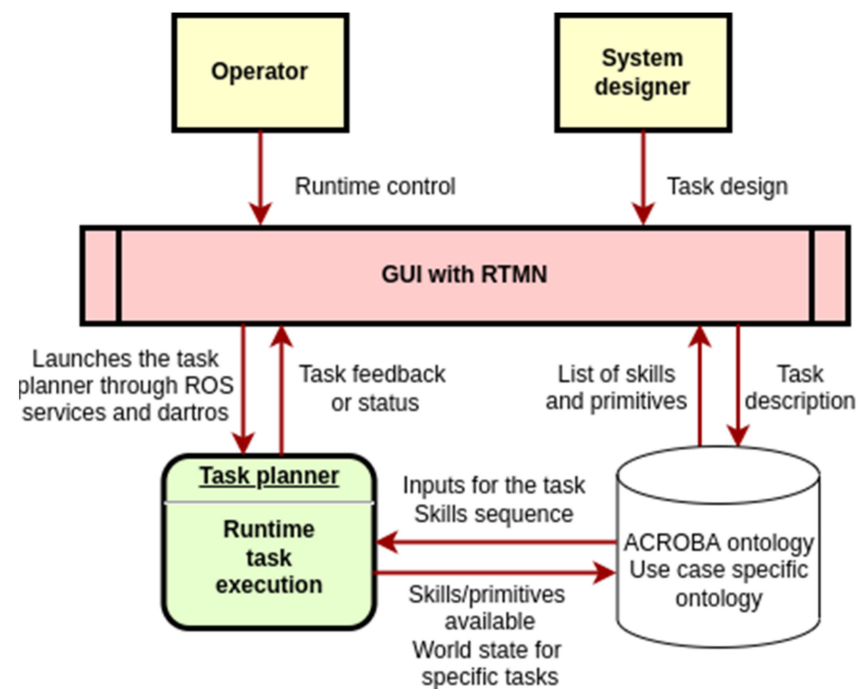

Figure 22. Primitive notation.

Figure 23. RTMN 2.0 framework [2].

The GUI using RTMN 2.0 serves as a modeling tool. The users can create a process model in the GUI. The process sequence is then sent from the GUI to the task planner through Dartros. We described the task planner in another research paper [59]. The task planner takes the sequence and "translates" it to a behavior tree, which is connected to ROS. The task planner also provides feedback and the status of the tasks to the GUI. The amount of available modeling elements, such as skills and primitives, is managed through an ontology where domain knowledge such as tasks, skills, and primitives are defined. The characteristics of ontology ease the sharing of knowledge, provide clearly structured data, and enable accurate reasoning of the data [60]. Therefore, we also adopted the benefits of ontology for knowledge sharing among heterogeneous systems and human–machine interactions.

2.2.7. Demonstration

The design of the Graphical User Interface (GUI) for RTMN 2.0 is presented in Figure 24. The left side is divided into different abstraction levels: processes, tasks, skills, and primitives. At the bottom left conner, the connecting notations are shown. One can simply drag and drop the notations on the left to model an HRC process.

We illustrated one example of an HRC process. This example is a modification of a use case process due to confidentiality. A printed circuit board (PCB) assembly is a process used to assemble a PCB. The assembly works by putting different kinds of Pin through Hole (PTH) components onto the PCB. This process consists of three tasks: move PTH to

pick station, assemble PCB, and inspect PCB. Move PTH to pick station and assemble PCB are two HRC tasks, while inspect PCB is a robot task.
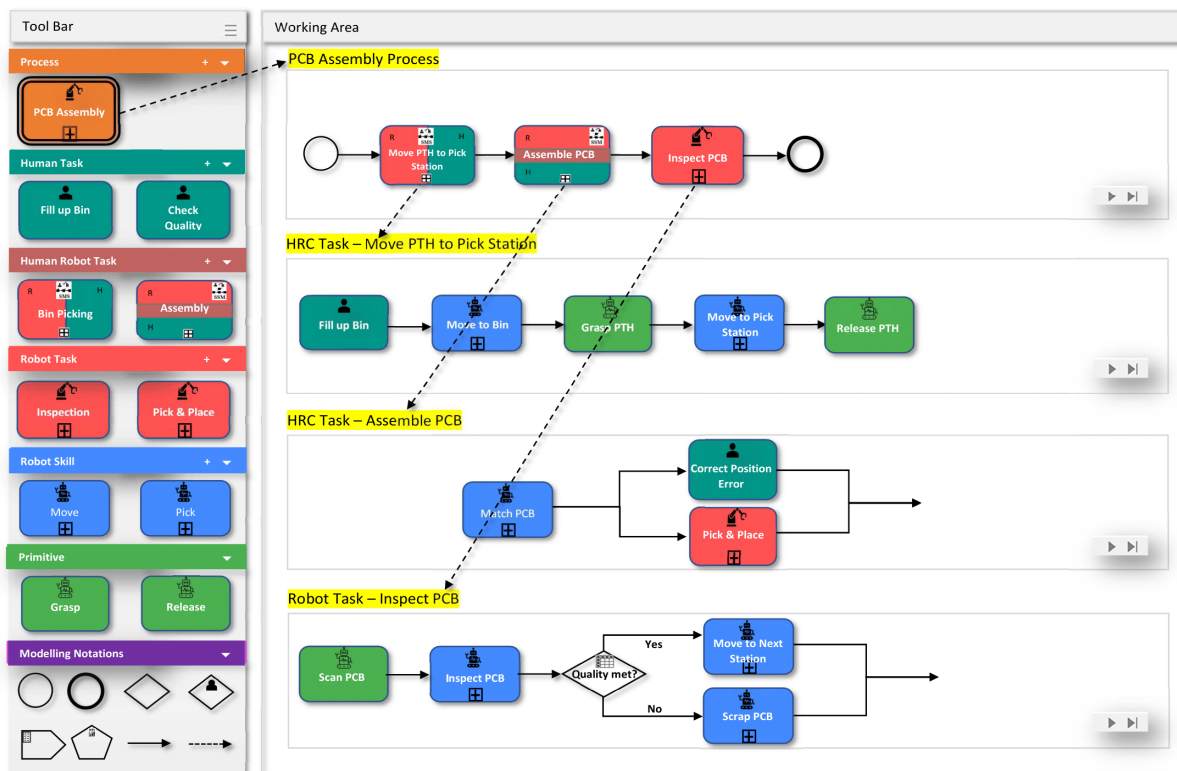

Figure 24. HRC process example—PCB assembly.

For the HRC task of move PTH to pick station, safety mode SMS should be ensured because it is a sequential HRC task with the human and robot working one after another. It has a human task, fill up bin; followed by a robot skill, move to bin; then a primitive grasp PTH; a robot skill, move to pick station; and a primitive, release PTH.

For the HRC task assemble PCB, safety mode SSM should be ensured, as humans and robots work in parallel for this task. They work side by side on the same PCB, but not at the same time. When a PTH is wrongly positioned, the human needs to correct it. As the human approaches the robot and enters the "red" zone, the robot stops. This task consists of a simple robot skill—match PCB (this skill scans PCB, then, based on the scanned image, loads the CAD model of the PCB)—and two parallel tasks (combination of skills)—correct position error by human and pick and place by the robot. The join/split notation is used here to split the sequence into two then merge the two back into one again.

The robot task inspect PCB does not involve any humans. It is purely performed by a robot. For this task, a condition notation is used to decide where the PCB goes after inspection. If it is accepted, it is moved to the next station. If it is rejected, the PCB is scrapped.

3. Results

We adopted the "technology acceptance model" [61] from Davis and Venkatesh for the purpose of developing and validating our approach. Based on Davis and Venkatesh, perceived ease of use and perceived usefulness of prototypes were shown to predict the real use of the technology [61]. We used both quantitative (questionnaire) and qualitative (interviews) methods. The evaluations of our approach were carried out with 17 partners and 82 participants.

As we described in our former paper [2], we first identified a gap between technology and users in the research area, which is "*Robot control requires high levels of expertise and*

operators lack knowledge and know-how for controlling robot". Keeping this gap in mind, we conducted a survey to gather requirements from the use cases. Based on the survey's findings, we developed the first version of the user interface. We performed an evaluation of this user interface through interviews and questionnaires. The main findings are described in the following sections.

3.1. Requirement Survey Findings

The survey is described in Appendix A. This survey/questionnaire received 16 answers. We illustrate the three key results gathered from the survey that are the most relevant for developing RTMN—see Figures 24–26.
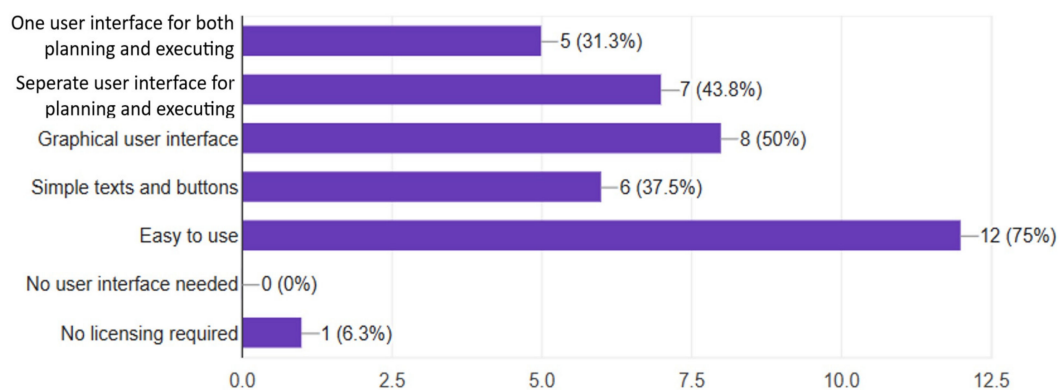
Figure 25. Survey Question 5: What do you expect for a user interface of ACROBA?
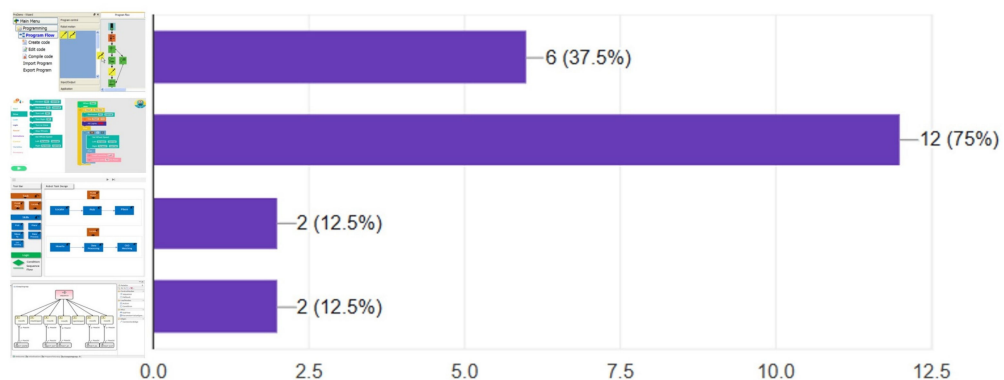
Figure 26. Survey Question 5: Which of the following task representations do you like the most?

The most important factors of a user interface are "Easy to use" and "graphical", as shown in Figure 25.

The RTMN representation received 12 votes and was the users' favorite choice (see Appendix A for question detail) (See Figure 26).

Figure 27 shows that "drag and drop" and "user friendly" are the most important requirements for a user interface.

To summarize, the main requirements identified by the users and implemented in the RTMN design for the user interface are that it is easy to use, graphical, drag and drop, and user friendly.

16 responses
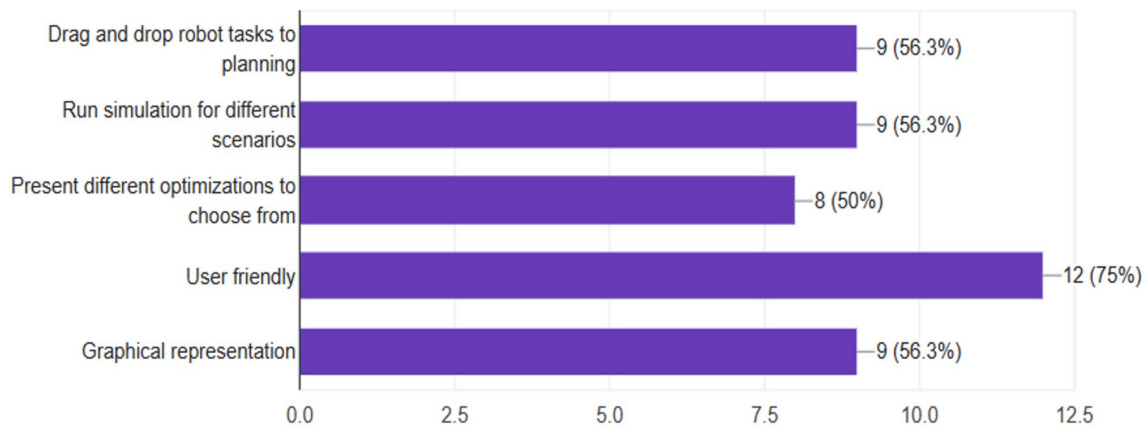
Figure 27. Survey Question 6: What do you expect from a user interface for planning the robot tasks?

3.2. Early Validation Interview and Survey Findings

The validation was carried out by both interviews and questionnaires. We interviewed seven user groups from five use cases of ACROBA and showed them a video of the first version of the graphical user interface with RTMN. The main results of the validation are shown in Figure 28 (not all questions are listed).
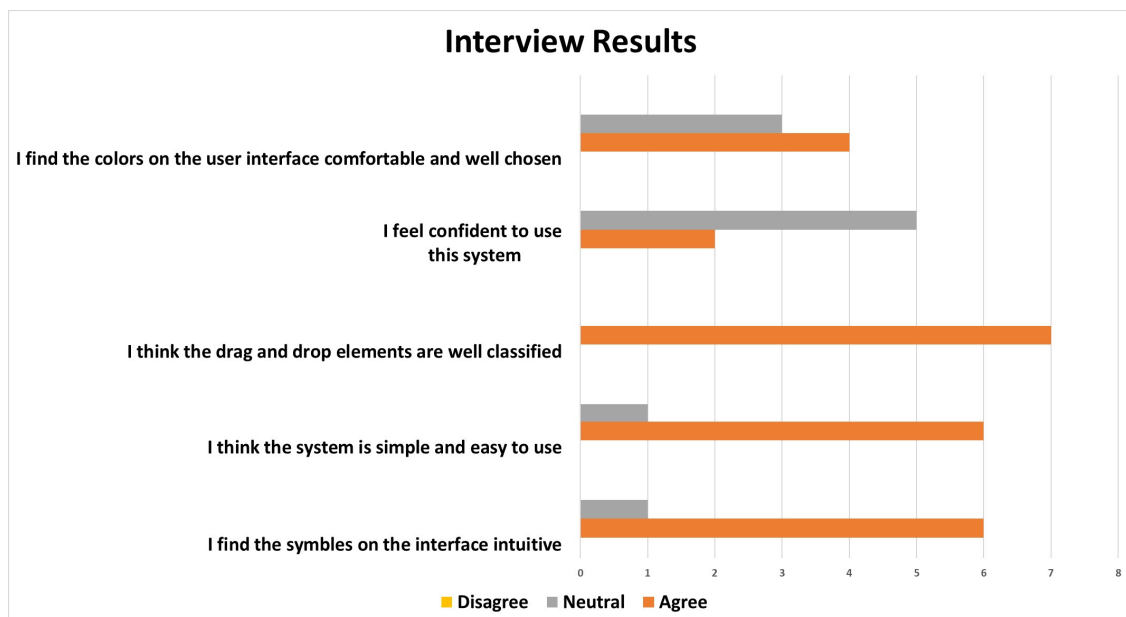
Figure 28. Interview results.

One can clearly see that they perceived the GUI with RTMN as useful for managing their work and expressed their willingness to use the tool. Additionally, they found the design easy to understand and intuitive to use. Therefore, based on the technology acceptance models [62], we can draw the conclusion that the likelihood for a user to use the tool after implementation is high, and the acceptance of the tool is promising.

Many additional requirements have arisen from the interviews. Human interaction with the robot, user input, and parallel process are the major areas of focus for the next version of the GUI. We gathered these requirements and converted them into new functional specifications for the development of the second version of the GUI. These functions are described in this paper.

We plan to perform another round of evaluation of the GUI after its implementation in mid-2024. We will again use both quantitative and qualitative research methods. We will analyze the results of the questionnaire and interviews and assess the benefits and drawbacks of the GUI. Based on the feedback, further improvements will be made.

There is a lengthy validation process after rollout to the five real-world industrial use cases. This will validate cost saving, productivity improvements, safety assurance, and efficiency. We have two additional third-party use cases to evaluate the flexibility and scalability of the tool. The results will be presented in 2024.

4. Discussion

This paper has not yet covered the details of modeling of human factors related to mental safety (mental stress and anxiety) induced by close interaction with robots. According to Hancock et al., establishing trust and acceptance is crucial in automation [63]. Norman emphasized user-centered design and its pivotal role [64]. He explained that poorly designed interfaces contribute to stress. Wickens et al., in their human factors engineering research, found out that effective workload management is essential to prevent cognitive and physical overload [65]. Asaro worked on the ethical and psychological impact of interacting closely with robots, and he described that these closed interactions may raise ethical concerns and have psychological implications, such as fear or discomfort [66].

Although challenges related to mental safety in HRC have been discovered, there are still many difficulties when addressing these changes: 1. collaboration and coordination between experts across various areas, such as robotics, psychology, human–computer interaction, and ethics; 2. the difficulty of keeping up with rapid technological advancements; 3. different people act differently when working with robots; and 4. we do not have a complete understanding of human cognition and emotion.

The authors will continue their research in this area and aim to develop a comprehensive approach for modeling human factors. Our plan is to address the following factors in future research: performance factors, such as productivity, capability, intension, and workload, and health factors, such as fatigue, anxiety, satisfaction, trust, and acceptance. We will define methods for assessment of these factors using system knowledge and facts saved in an ontology. This assessment can be defined in the form of rules for the HRC tasks. Based on the conditions of the rules, actions will be triggered for task planning and execution.

5. Conclusions

This paper covers the extension and modification of the RTMN. RTMN 2.0 focuses on the modeling of human robot collaboration processes. Modeling of the requirements and KPIs was added to enable traceability from a process to the lower robotic control level. Notations related to decision making are modified as an improvement to RTMN. These additional features of RTMN 2.0 together extend RTMN to the HRC process domain by separating the HRC tasks from normal tasks (human tasks and robot tasks) and labeling the different types of human–robot interactions. The five separation notations are self-explanatory, and users can use them as predefined templates, to easily design the task details, and to ensure safety through safety modes, which require configuration for hardware and software implementations. The safety of humans is ensured through the assigned safety mode of each notation, as each safety mode is monitored in the background with corresponding rules to control the safety. These additional features provide the following benefits to users: increases the intuitiveness, provides easy-to-use templates, is code-free, and ensures safety and reusability. In addition, the basic RTMN notations are very similar to the notations users use to draw process flows. The familiarity then eases the utilization of the tool, therefore allowing people without programming expertise to plan and control robots in an easier way.

User training is planned before the final testing after the implementation. This training is expected to be within 10 h. The simplicity of the tool is very high in comparison to gaining high-level robot programming skills, which would take months or years.

## Appendix A. Questionnaire for ACROBA User Interface (Plain Version, Link to Google form: https://forms.gle/6iyfGuF2giEUWMWX7)

Role_____, Expert Level_____, Relation to the project_____

1. What are the users of the ACROBA platform (allow multiple choice)?

    A. Business users
    B. Engineers
    C. Programmers
    D. Manager
    E. Others: _____

2. Do the users want to interact with the system?

    A. Yes
    B. No
    C. Only when error occurs.
    D. Others: _____

3. On what device do you expect to manage the control of the robot?

    A. Smartphone/tablet
    B. Laptop/PC
    C. HMI
    D. Others: _____

4. What do you expect for a user interface of ACROBA?

    A. One user interface for both planning and execution
    B. Separate user interface for planning and execution
    C. Graphical user interface
    D. Simple texts and buttons
    E. Easy to use.
    F. No user interface needed.

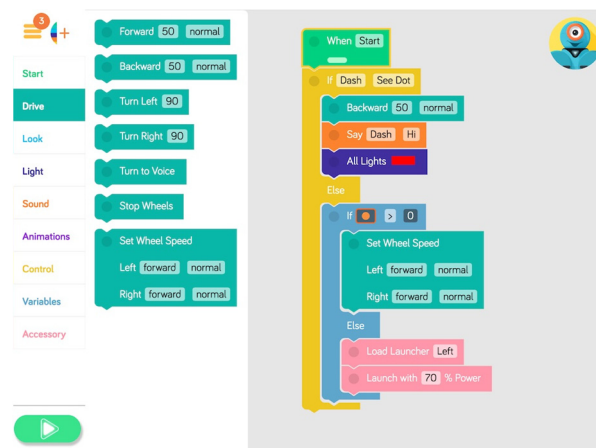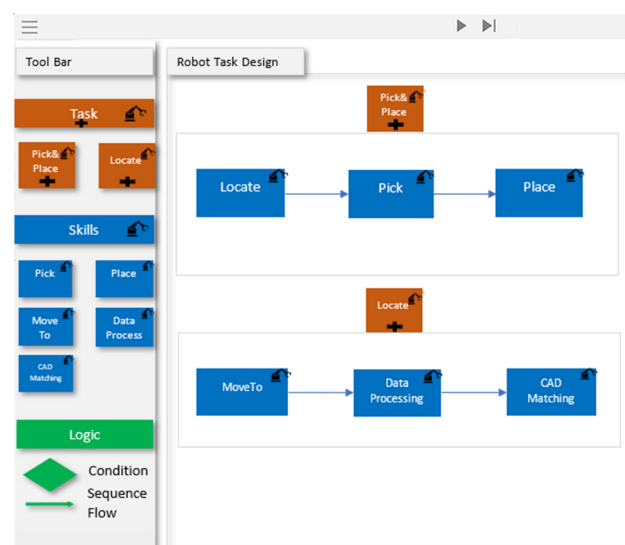5. Which of the following task representations do you like the most?

a.
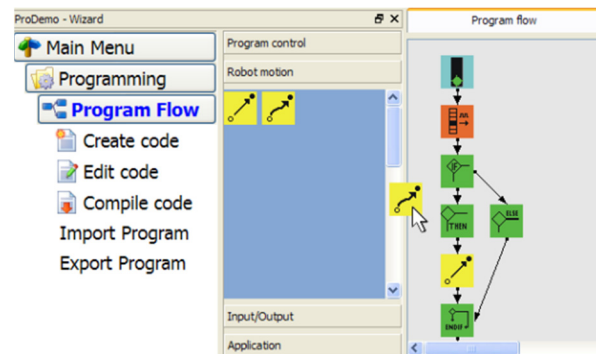
Source: https://www.researchgate.net/figure/sual-Programming-using-Drag-and-Drop-to-assemble-the-program-flow_fig6_226029194, accessed on 12 November 2023.
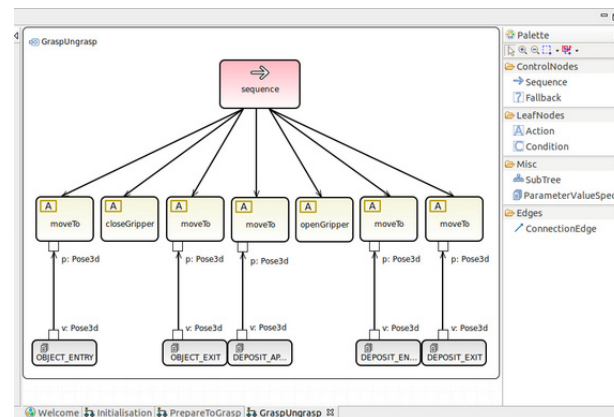
b.

Source: https://home.makewonder.com/apps/blockly, accessed on 12 November 2023.

c.

Source: RTMN

d.

Scheme 12. November 2023.

6 What do you expect from a user interface for planning robot tasks?

    A.   Drag and drop robot tasks to planning.
    B.   Run simulation for different scenarios.
    C.   Present different optimizations to choose from
    D.   User friendly
    E.   Graphical representation
    F.   Others: _____

7 What do you expect from a user interface when executing robot tasks?

    A.   An overview of the task execution
    B.   Control the robot tasks execution: _____

        a.   go forward.
        b.   backward
        c.   stop
        d.   repeat
        e.   run optimization.
        f.   Others: _____

    C.   Provide additional information from different systems when error occurs.

        a.   Information from the Vision system
        b.   Information from the safety system
        c.   Information from robot task
        d.   Others: _____

8 What do you want to control robot tasks execution?

    a.   go forward.
    b.   go backward
    c.   stop
    d.   repeat
    e.   run optimization.
    f.   Others: _____

9 What information do you want to see when error occurs?

    a.   Information from the Vision system
    b.   Information from the safety system
    c.   Information from robot task
    d.   Others: _____

10 Do you have experience working with Robot in the production line?

    A.   Yes
    B.   No

11   If yes, what are the problems you have working with robots?

    A.   When there is an error, it is hard to fix the error.

    B.   Error/warning are too technical, not enough information of the system, cannot fix the problem easily.

    C.   Cannot control the robot tasks freely.

    D.   Reprogramming a robot task is very time consuming.

    E.   No overview of the process

    F.   Lacking information from the vision system or other systems of the tasks performed

    G.   User interface is too simple.

    H.   The user interface is not user friendly.

    I.   Others: _____

References

1. Project—Acroba Project. Available online: https://acrobaproject.eu/project-acroba/ (accessed on 28 February 2022).
2. Sprenger, C.Z.; Ribeaud, T. Robotic Process Automation with Ontology-Enabled Skill-Based Robot Task Model and Notation (RTMN). In Proceedings of the 2nd International Conference on Robotics, Automation and Artificial Intelligence (RAAI), Singapore, 9–11 December 2022; pp. 15–20. [CrossRef]
3. BPMN Specification—Business Process Model and Notation. Available online: https://www.bpmn.org/ (accessed on 27 February 2022).
4. Li, Y.; Ge, S.S. Human-Robot Collaboration Based on Motion Intention Estimation. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 1007–1014. [CrossRef]
5. Weiss, A.; Wortmeier, A.K.; Kubicek, B. Cobots in Industry 4.0: A Roadmap for Future Practice Studies on Human-Robot Collaboration. *IEEE Trans. Hum. Mach. Syst.* **2021**, *51*, 335–345. [CrossRef]
6. Lubold, N.; Walker, E.; Pon-Barry, H. Effects of voice-adaptation and social dialogue on perceptions of a robotic learning companion. In Proceedings of the HRI'16: The 11th ACM/IEEE International Conference on Human Robot Interation, Christchurch, NZ, USA, 7–10 March 2016.
7. Institute of Electrical and Electronics Engineers. *A Special Project of the IEEE Region 3 Strategic Planning Committee*; IEEE: Piscataway, NJ, USA, 2015.
8. Freedy, A.; DeVisser, E.; Weltman, G.; Coeyman, N. Measurement of Trust in Human-Robot Collaboration. In Proceedings of the International Symposium on Collaborative Technologies and Systems, Orlando, FL, USA, 25 May 2007.
9. IEEE Staff. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*; IEEE: Piscataway, NJ, USA, 2010.
10. Kumar, S.; Savur, C.; Sahin, F. Survey of Human-Robot Collaboration in Industrial Settings: Awareness, Intelligence, and Compliance. *IEEE Trans. Syst. Man. Cybern. Syst.* **2021**, *51*, 280–297. [CrossRef]
11. Kock, S.; Vittor, T.; Matthias, B.; Jerregard, H.; Källman, M.; Lundberg, I.; Hedelind, M. Robot concept for scalable, flexible assembly automation: A technology study on a harmless dual-armed robot. In Proceedings of the IEEE International Symposium on Assembly and Manufacturing, Tampere, Finland, 25–27 May 2011.
12. Matheson, E.; Minto, R.; Zampieri, E.G.G.; Faccio, M.; Rosati, G. Human–Robot Collaboration in Manufacturing Applications: A Review. *Robotics* **2019**, *8*, 100. [CrossRef]
13. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on Human-Robot Collaboration in Industrial Settings: Safety, Intuitive Interfaces and Applications. *Mechatronics* **2018**, *55*, 248–266. [CrossRef]
14. Kim, A. A Shortage of Skilled Workers Threatens Manufacturing's Rebound. Available online: https://www.ge.com/news/reports/a-shortage-of-skilled-workers-threatens-manufacturings-r (accessed on 23 October 2023).
15. Vysocky, A.; Novak, P. Human—Robot Collaboration in Industry. *MM Sci. J.* **2016**, 903–906. [CrossRef]
16. International Federation of Robotics. Available online: https://ifr.org/ (accessed on 18 July 2023).
17. *ISO 12100:2010*; Safety of Machinery—General Principles for Design—Risk Assessment and Risk Reduction. International Organization for Standardization: Geneva, Switzerland, 2010.
18. *ISO 13849-1:2023*; Safety-Related Parts of Control Systems—Part 1: General Principles for Design. International Organization for Standardization: Geneva, Switzerland, 2023.
19. *ISO 13850:2015*; Safety of Machinery—Emergency Stop Function—Principles for Design. International Organization for Standardization: Geneva, Switzerland, 2015.
20. *ISO 13851:2019*; Safety of Machinery—Two-Hand Control Devices—Principles for Design and Selection. International Organization for Standardization: Geneva, Switzerland, 2019.
21. *ISO 10218-1:2011*; Robots and Robotic Devices—Safety Requirements for Industrial Robots—Part 1: Robots. International Organization for Standardization: Geneva, Switzerland, 2011.
22. *ISO 10218-2:2011*; Robots and Robotic Devices—Safety Requirements for Industrial Robots—Part 2: Robot Systems and Integration. International Organization for Standardization: Geneva, Switzerland, 2011.

23. *EC 62061*; Safety of Machinery Functional Safety of Safety-Related Electrical, Electronic and Programmable Electronic Control System. International Electrotechnical Commission: London, UK, 2005.

24. *ISO/TS 15066:2016*; Robots and Robotic Devices—Collaborative Robots. International Organization for Standardization: Geneva, Switzerland, 2016.

25. Müller, R.; Vette, M.; Geenen, A. Skill-Based Dynamic Task Allocation in Human-Robot-Cooperation with the Example of Welding Application. *Procedia Manuf.* **2017**, *11*, 13–21. [CrossRef]

26. Wang, L.; Gao, R.; Váncza, J.; Krüger, J.; Wang, X.V.; Makris, S.; Chryssolouris, G. Symbiotic Human-Robot Collaborative Assembly. *CIRP Ann.* **2019**, *68*, 701–726. [CrossRef]

27. Thiemermann, S. Direkte Mensch-Roboter-Kooperation in Der Kleinteilemontage Mit Einem SCARA-Roboter. Ph.D. Thesis, University of Stuttgart, Stuttgart, Germany, 2004.

28. Müller, R.; Vette, M.; Mailahn, O. Process-Oriented Task Assignment for Assembly Processes with Human-Robot Interaction. *Procedia CIRP* **2016**, *44*, 210–215. [CrossRef]

29. Vincent Wang, X.; Kemény, Z.; Váncza, J.; Wang, L. Human-Robot Collaborative Assembly in Cyber-Physical Production: Classification Framework and Implementation. *CIRP Ann.* **2017**, *66*, 5–8. [CrossRef]

30. Krü, J.; Lien, T.K.; Verl, A. Cooperation of Human and Machines in Assembly Lines. *CIRP Ann.* **2009**, *58*, 628–646. [CrossRef]

31. White, S.A. *Introduction to BPMN*; Ibm Cooperation: New York, NY, USA, 2004.

32. Lindorfer, R.; Froschauer, R. Towards user-oriented programming of skill-based Automation Systems using a domain-specific Meta-Modeling Approach. In Proceedings of the IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, 22–25 July 2019; pp. 655–660. [CrossRef]

33. Pantano, M.; Pavlovskyi, Y.; Schulenburg, E.; Traganos, K.; Ahmadi, S.; Regulin, D.; Lee, D.; Saenz, J. Novel Approach Using Risk Analysis Component to Continuously Update Collaborative Robotics Applications in the Smart, Connected Factory Model. *Appl. Sci.* **2022**, *12*, 5639. [CrossRef]

34. Mahulea, C.; Grau, A.; Lo Bello, L. The 24th IEEE International Conference on Emerging Technologies and Factory Automation Held in Zaragoza, Spain [Society News]. *IEEE Ind. Electron. Mag.* **2019**, *13*, 127–128. [CrossRef]

35. Schmidbauer, C.; Schlund, S.; Ionescu, T.B.; Hader, B. Adaptive Task Sharing in Human-Robot Interaction in Assembly. In Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 14 December 2020; pp. 546–550.

36. Guiochet, J. Hazard Analysis of Human-Robot Interactions with HAZOP-UML. *Saf. Sci.* **2016**, *84*, 225–237. [CrossRef]

37. Martin-Guillerez, D.; Guiochet, J.; Powell, D.; Zanon, C. A UML-Based Method for Risk Analysis of Human-Robot Interactions. In Proceedings of the 2nd International Workshop on Software Engineering for Resilient Systems, New York, NY, USA, 15 April 2010; pp. 32–41.

38. Guiochet, J.; Motet, G.; Baron, C.; Boy, G. Toward a human-centered uml for risk analysis application to a medical robot. In *Human Error, Safety and Systems Development*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 177–191.

39. Guiochet, J.; Hoang, Q.A.D.; Kaaniche, M.; Powell, D. Model-based safety analysis of human-robot interactions: The MIRAS walking assistance robot. In Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR), Seattle, WA, USA, 24–26 June 2013.

40. Von Borstel, F.D.; Villa-Medina, J.F.; Gutiérrez, J. Development of Mobile Robots Based on Wireless Robotic Components Using UML and Hierarchical Colored Petri Nets. *J. Intell. Robot. Syst. Theory Appl.* **2022**, *104*, 70. [CrossRef]

41. Carroll, L.; Tondu, B.; Baron, C.; Geffroy, J.C. *UML Framework for the Design of Real-Time Robot Controllers*; Springer: Berlin/Heidelberg, Germany, 1999.

42. Verband der Elektrotechnik, E.; Institute of Electrical and Electronics Engineers. Modeling Robot Assembly Tasks in Manufacturing Using SysML. In Proceedings of the 41st International Symposium on Robotics, Munich, Germany, 2–3 June 2014.

43. Ohara, K.; Takubo, T.; Mae, Y.; Arai, T. SysML-Based Robot System Design for Manipulation Tasks. In Proceedings of the 5th International Conference on the Advanced Mechatronics, Shenzhen, China, 18–21 December 2020.

44. Candell, R.; Kashef, M.; Liu, Y.; Foufou, S. A SysML Representation of the Wireless Factory Work Cell: Enabling Real-Time Observation and Control by Modeling Significant Architecture, Components, and Information Flows. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 119–140. [CrossRef] [PubMed]

45. Avram, O.; Baraldo, S.; Valente, A. Generalized Behavior Framework for Mobile Robots Teaming with Humans in Harsh Environments. *Front. Robot. AI* **2022**, *9*, 898366. [CrossRef] [PubMed]

46. Peterson, J.L. Petri Nets*. *Comput. Surv.* **1977**, *9*, 223–252. [CrossRef]

47. Casalino, A.; Zanchettin, A.M.; Piroddi, L.; Rocco, P. Optimal Scheduling of Human-Robot Collaborative Assembly Operations with Time Petri Nets. *IEEE Trans. Autom. Sci. Eng.* **2019**, *18*, 70–84. [CrossRef]

48. Chao, C.; Thomaz, A. Timed Petri Nets for Fluent Turn-Taking over Multimodal Interaction Resources in Human-Robot Collaboration. *Int. J. Rob. Res.* **2016**, *35*, 1330–1353. [CrossRef]

49. Chao, C.; Thomaz, A. Timing in Multimodal Turn-Taking Interactions: Control and Analysis Using Timed Petri Nets. *J. Hum. Robot. Interact.* **2012**, *1*, 4–25. [CrossRef]

50. Institute of Electrical and Electronics Engineers. *Proceedings of the ICRA 2014—IEEE International Conference on Robotics and Automation*; IEEE: Piscataway, NJ, USA, 2014.

51. Yagoda, R.E.; Coovert, M.D. How to Work and Play with Robots: An Approach to Modeling Human-Robot Interaction. *Comput. Hum. Behav.* **2012**, *28*, 60–68. [CrossRef]

52. Casalino, A.; Cividini, F.; Zanchettin, A.M.; Piroddi, L.; Rocco, P. *Human-Robot Collaborative Assembly: A Use-Case Application*; Elsevier B.V.: Amsterdam, The Netherlands, 2018; Volume 51, pp. 194–199.

53. Völzer, H. An Overview of BPMN 2.0 and Its Potential Use. In *Business Process Modeling Notation*; Lecture Notes in Business Information Processing; Springer: Berlin/Heidelberg, Germany, 2010; Volume 67, pp. 14–15. [CrossRef]

54. ISO—International Organization for Standardization. Available online: https://www.iso.org/obp/ui/en/#iso:std:iso:10218:-1:ed-2:v1:en (accessed on 7 November 2023).

55. Caiazzo, C.; Nestić, S.; Savković, M. A Systematic Classification of Key Performance Indicators in Human-Robot Collaboration. *Lect. Notes Netw. Syst.* **2023**, *562*, 479–489.

56. Flutter. Available online: https://flutter.dev/ (accessed on 15 August 2022).

57. Dartros | Dart Package. Available online: https://pub.dev/packages/dartros (accessed on 15 August 2022).

58. ROS: Home. Available online: https://www.ros.org/ (accessed on 28 February 2022).

59. Ribeaud, T.; Sprenger, C.Z. Behavior Trees Based Flexible Task Planner Built on ROS2 Framework. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Stuttgart, Germany, 6–9 September 2022. [CrossRef]

60. Pang, W.; Gu, W.; Li, H. *Ontology-Based Task Planning for Autonomous Unmanned System: Framework and Principle*; Journal of Physics: Conference Series; IOP Publishing Ltd.: Bristol, UK, 2022; Volume 2253.

61. Davis, F.D.; Venkatesh, V. Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management. *IEEE Trans. Eng. Manag.* **2004**, *51*, 31–46. [CrossRef]

62. Davis, F.D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *Manag. Inf. Syst. Q.* **1989**, *13*, 319–339. [CrossRef]

63. Hancock, P.A.; Billings, D.R.; Schaefer, K.E. Can You Trust Your Robot? *Ergon. Des. Q. Hum. Factors Appl.* **2011**, *19*, 24–29. [CrossRef]

64. Norman, D. *The Design of Everyday Thing*; Basic Books: New York, NY, USA, 2013.

65. Wickens, C.D.; Gordon, S.E.; Liu, Y.; Lee, J. *An Introduction to Human Factors Engineering*; Pearson Prentice Hall: Upper Saddle River, NJ, USA, 2004; Volume 2.

66. Asaro, P.M. What Should We Want from a Robot Ethic? In *Machine Ethics and Robot Ethics*; Routledge: London, UK, 2020; pp. 87–94.