2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA) | 979-8-3315-5383-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/ETFA65518.2025.11205620

Using Behavior Trees in Risk Assessment

Razan Ghzouli*, Atieh Hanna†, Endre Erös‡, and Rebekka Wohlrab*§
*Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden
†Volvo Group, Gothenburg, Sweden
‡Chalmers Industriteknik, Gothenburg, Sweden
§Carnegie Mellon University, Pittsburgh, USA

***Abstract*—Cyber-physical production systems increasingly involve collaborative robotic missions, which come with a higher demand for robustness and safety. Practitioners rely on risk assessments to identify potential failures and implement measures to mitigate their risks. Ensuring that mitigation strategies derived from risk assessments are adequately considered in the software implementation can be challenging, especially when stakeholders involved in the assessment process lack a programming background. This leads to a disconnection between the outputs of risk assessments and the actual implementation of robotic missions. To address this issue, there is a need to integrate software engineering practices into the risk assessment process to ensure consistency and traceability between the outputs of risk assessments and their corresponding software implementation.**

This paper presents a design science study that conceived a model-based approach for early risk assessment in a development-centric way. Our approach supports risk assessment activities by using behavior-tree models. We evaluated the approach together with five practitioners from four companies. This approach is the first attempt to use behavior-tree models to support risk assessment. Our findings highlight the potential of behavior-tree models in supporting early identification, visualization, and bridging the gap between code implementation and the outputs of risk assessments. Our findings suggest research directions for further development of the approach to increase its applicability and usefulness in practice.

***Index Terms*—behavior trees, risk assessment, model-based engineering, safety, robotics, design science**

I. Introduction

Cyber-Physical Production Systems (CPPSs) increasingly involve collaborative settings where humans and robots work side by side. Safety and robustness are crucial properties that need to be guaranteed in robotic missions and systems, especially in collaborative missions. In a large-scale empirical study on the state of robotics software engineering, the robustness of robotics systems was ranked as the most pressing challenge in practice [1].

Risk assessment is the first step of safety analysis, which aims to identify different hazards that can lead to system failure [2]. In the context of robotic missions, when an action fails, it can lead to reduced performance or even more catastrophic safety risks. Not capturing failures early in the development of robotics systems may lead to technical debt, making future modifications and maintenance more complex and expensive [3]. However, as with any non-code artifacts, there is a risk that practitioners do not see the value of risk assessments if they are disconnected from the actual implementation of robotic missions [4]. To facilitate their practical adoption, it is crucial to ensure that the outputs of risk assessments are well-integrated and transferred into the code implementation of robotics missions. Currently, there is a gap between safety analysis and implementation artifacts, making it difficult for practitioners to see the value of risk assessments [5]. This paper aims to address these challenges by presenting a model-based approach that supports early risk assessment in a development-centric way.

In the software engineering community, various modeling approaches have been conceived to support the development of robotic missions and systems. While modeling approaches are not always adopted in practice, one approach that has gained increasing popularity among robotics practitioners is behavior trees. Behavior trees model and execute the actions and control-switching mechanisms involved in a robotic mission. Key benefits of behavior trees are their understandability [6] and their support to account for reactiveness, by modeling fallback behaviors in case an action cannot be successfully completed [7]. Due to their popularity, it appears promising to investigate the use of behavior trees for supporting risk assessment. To the best of our knowledge, there is no previous research on using behavior trees for safety risk assessment.

This paper presents a design science study that aims to conceive a model-based approach for early risk assessment in a development-centric way. We iteratively aimed to understand current problems, develop the model-based risk assessment approach, and evaluate it with five practitioners. Our approach integrates behavior trees into the risk assessment process to overcome practical challenges. We found that using an explicit mission model at the early design stages of risk assessment can enhance comprehension of the mission and identification of possible failures. Our findings highlight the potential of behavior trees for holding and documenting the outputs of the risk assessment used, as well as aligning the implementation code with assessment outputs. Our findings further demonstrate the challenge of determining the optimal granularity level of behavior-tree models, and the necessity of an automated approach and improved tooling to transfer assessment outputs. We see our findings as a promising first step, opening up new directions for future research in using behavior trees in risk assessments. All our data is available at an accompanying online appendix [8].

II. Background and Related Work

A. Background

Behavior Trees. Behavior trees are modular and flexible, providing intuitive and understandable models for robotics

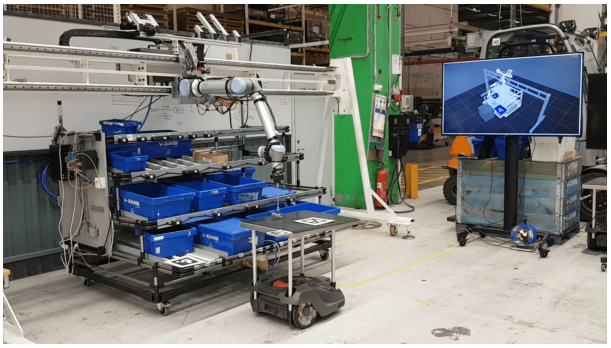

Fig. 1: The collaborative robotic system (Robot in the Air).

missions [7]. Behavior-tree models are directed tree structures containing a root node, control-flow nodes (non-leaf nodes), and execution nodes (leaf nodes). Execution proceeds via "ticks" originating from the root and traverses the tree based on the semantics of the control-flow nodes. Control-flow nodes orchestrate the execution of child nodes, enabling the coordination of complex tasks. The four main types of control-flow nodes are sequence, selector, decorator, and parallel nodes. Execution nodes are where the actual execution of the code is defined. They are either robotic actions or conditions evaluating propositions. Each ticked node returns a status (success, failure, or running) to its parent, facilitating dynamic behavior adaptation.

`BehaviorTree.CPP`[1] is the C++ implementation of behavior trees, and one of the most used libraries in open-source projects [9]. `BehaviorTree.CPP` has a graphical user interface called Groot that supports the creation and monitoring of behavior trees. We direct interested readers to previous works for more information on available languages for behavior trees [9].

Risk Assessments. Risk assessments are mandatory for most robotics applications in industrial production within an EU regulatory context. They could be done in correspondence with the Machinery Directive (2006/42/EC) [10] or according to the robotic harmonized standards such as ISO 12100 [11]. Risk assessment typically begins with hazard identification, estimation, and analysis of their severity, which ends with risk reduction. However, the above standards lack guidelines for assessing intelligent and collaborative robotics systems. Risk assessments are criticized in practice for being time-consuming [12].

Failure mode effect analysis (FMEA), well established in the automotive industry [13], is an engineering analysis method based on reliability theory [14]. It assesses systems, processes, or software by evaluating potential failures, their effects and causes, and determining current controls to prevent or detect failures. FMEA employs a bottom-up approach to identify failure modes and causes in an easy and structured manner.

The Collaborative Robotics Mission. We investigate using a collaborative robotic system called Robot in the Air (RITA), an intelligent automation system for assembling kits in an assembly line. RITA is a pick-to-light system, which is a paperless picking system with a light-directed picking [15]. Figure 1 shows RITA, which is composed of a gantry, a mounted robotic arm equipped with a scanner, and an autonomous trolley. The scanner detects and localizes items to be picked from toolboxes (blue boxes in Fig. 1). The system should allow the operators and robots to work in a shared zone. The system can perform tasks by picking ordered items and putting them on the autonomous trolley to bring the materials to an assembly station. Controlling such a system involves various challenges, including but not limited to ensuring the precision and reliability of scanning and localization of items, preventing collisions between the robot and operators or surrounding equipment, quickly and adequately responding to failures, and achieving a balance between execution speed to meet assembly cycle deadlines and ensuring operator safety and equipment protection.

[1]https://github.com/BehaviorTree/BehaviorTree.CPP

B. Related Work

Bdiwi et al. [12] and Abdulkhaleq and Wagner [16] proposed integrating state machines into risk assessments, which are another type of behavioral model. Bdiwi et al. used state machines for modeling risk mitigation strategies, and Abdulkhaleq and Wagner used them to provide appropriate diagrammatic notations to represent the safety control structure in systems theoretic process analysis (STPA). The aforementioned research highlights the importance of a clear visual behavior model illustrating the relationships between mission components and control actions to identify potential risks. Unlike our work, none of the previous works explored integrating behavior-tree models into risk assessment approaches.

Previous research [17], [18] shows the potential of behavior trees for modeling and executing mitigation strategies based on system constraints such as safety. In the framework proposed by Castano and Xu [18], behavior trees were used to model risk mitigation strategies by detecting failures and reacting to mitigate them. Although risk assessments are mentioned as an important component of their framework, they do not mention the risk assessment used or the process for mapping the risk assessment's outputs. Our work builds on this potential and advocates for incorporating behavior-tree models into the risk assessment approach from the early design phase of robotics projects, especially in industry, to improve the transferability and communication between stakeholders.

III. Research Method

We adopted the design science method [19] to create our model-based approach for early risk assessment. Two research questions guided our study:

RQ1. How can a model-based approach be designed to support risk assessments in a development-centric way?

Our goal was to investigate how we can support practitioners in robotics while doing risk assessments. As stated in the introduction, the approach relies on behavior trees, given their popularity, understandability, and support to model reactiveness.

To understand the requirements for such an approach, we conducted a workshop to understand the challenges faced by practitioners in an automotive company. Risk assessments

were identified as a challenging step in robotics projects. Through follow-up meetings at the company, we discussed using behavior trees to mitigate some of the challenges, and we collaborated with the involved practitioners to develop an approach to using behavior-tree models during the risk assessment of robotics missions.

RQ2. To what extent can our proposed approach support practitioners in performing risk assessments in a development-centric way?

Design science involves the iterative understanding of practical problems, the development of a design artifact (i.e., our risk assessment approach), and the evaluation of the artifact [19]. We performed two cycles and involved practitioners from several companies. In cycle 1, we relied on an internal evaluation with a developer who was not involved in the initial conception of the approach. In cycle 2, we included participants from external teams/companies to evaluate the approach.

A. Selected Companies and Participants

The proposed approach was developed with practitioners from an automotive manufacturer (Company A) with more than 104,000 employees, and a research and development organization (Company B) with more than 100 employees that works closely with Company A.

Internal Research Team: Throughout this paper, we use the term "internal participants" to refer to the internal research team. Two practitioners were involved during the whole study from Companies A and B. During the work, we divided ourselves into two teams to reduce bias.

Team 1 comprised the safety expert from Company A and a researcher from academia (the first and second authors). Team 1 was involved in creating our proposed approach and applying it to the collaborative robotic mission.

Team 2 is the industrial researcher from Company B (third author), who has technical experience with the system RITA. In the rest of the paper, we refer to him as "the developer" of the robotic system under consideration. Team 2 was involved in the initial internal evaluation of the process outputs.

Participants in cycle 2: We conducted a think-aloud study to evaluate our proposed approach with external participants. The participants were chosen based on previous experience with risk analysis, availability, and existing connections. Five participants from four different companies were involved. Table I provides an overview of the participants. P1 and P2 are from different teams in Company A, and P3 is from Company B, with no prior involvement in the study. P4 is a safety and security researcher from an independent research institute collaborating with universities, industry, and the public sector. P5 is a senior product security engineer at a medical equipment manufacturer. We use the term "external participants" to refer to the participants from cycle 2.

The external participants have diverse backgrounds and knowledge. Four participants have previous experience with industrial robotics, ranging from 2 to 10 years. P1, P2, and P3 have prior experience with the system RITA under study. None of the participants had worked with behavior trees before.

TABLE I: Overview of the external participants (cycle 2).

	Role	**Domain and company**	**Exp. with safety analysis**
P1	senior researcher	automotive (Comp. A)	5-10 years
P2	senior researcher	automotive (Comp. A)	5-10 years
P3	researcher	research and development (Comp. B)	0 years
P4	safety and security researcher	research and development (Comp. C)	2-5 years
P5	product security engineer	medical (Comp. D)	5-10 years

B. Understanding the Environment

The initial phase involved understanding the challenges and requirements for risk assessment. We started by conducting a workshop. The first author held a workshop with the internal participants following Säfsten and Gustavsson guidelines [20]. During the session, behavior trees were explained, and then we discussed challenges faced in robotics projects. The researcher took notes on a whiteboard and paper, and later analyzed them. Based on this analysis, we derived a list of requirements to overcome the challenges faced during risk assessments (cf. Section IV-A).

C. Developing the Approach

We developed the model-based approach based on the extracted requirements from the workshop. The researcher (first author) visited the company multiple times for a month to develop the approach collaboratively with the safety expert. The researcher took notes during the sessions, and the process of developing the approach was iterative.

D. Evaluation

Internal Evaluation. The goal of the internal evaluation in cycle 1 was to test whether an enriched behavior-tree model would support developers in understanding safety concerns and transferring them into the implementation of the robotic mission. The developer, Team 2, was involved in this evaluation. Team 1 handed the developer the enriched behavior-tree model with the FMEA results to implement (code) the mission. We chose an existing behavior-tree tool to visualize the enriched behavior-tree model, called Groot. The developer provided feedback to Team 1 to improve the level of information annotated in the behavior-tree model. The feedback led to the iterative refinement of our risk assessment approach.

External Evaluation. In cycle 2, we conducted a think-aloud study [21] to evaluate the usefulness of using behavior-tree models for risk assessments of robotics missions, specifically in identifying risks, transferring the outputs into the implementation phase, and visualizing the outputs.

We organized individual think-aloud sessions with the external participants, held by the same researcher (first author). Each session was an hour long and was recorded for analysis. The presentation began with a brief introduction by the researcher to the behavior-tree models and the RITA system. Informed consent was obtained from each participant. The participants were given a document describing a pick-and-place mission for the robotic arm and a behavior-tree model of

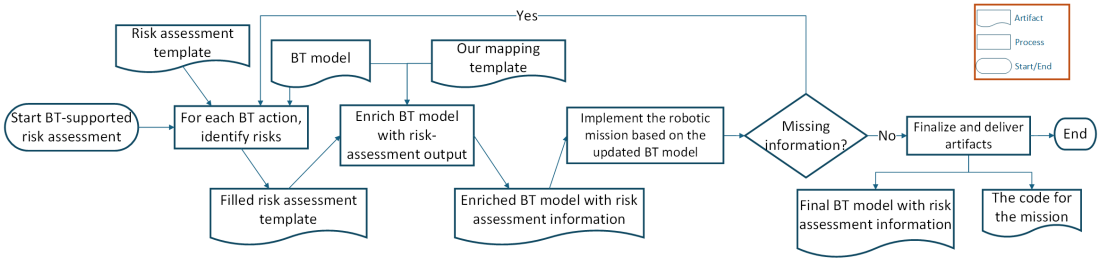

Fig. 2: Our model-based approach for supporting risk assessment with BTs.

the mission. The document included two tasks: (1) conduct an FMEA for the place part of the mission following our model-based approach, (2) transfer the outputs of the FMEA into a behavior-tree tool to help with the visualization of the outputs and the outputs' transfer into the implementation phase. We used Groot to facilitate the visualization and mapping of the outputs of the FMEA into the behavior-tree model. We also provided an Excel file to conduct the FMEA, along with a prefilled example for the pick part of the mission to guide participants, which can be found in our online appendix.

At the end of the sessions, we provided a three-part survey. The first part was the raw NASA Task Load Index (RTLX) [22] to assess the overall workload of our approach. The RTLX used a scale of 1 to 10 for each subscale, and no weighting was applied to the dimensions. The second part concerned reflections on the usefulness and the most favorable and the least favorable aspects of using the behavior-tree model during the given tasks. The final part addressed the future application of the entire approach, the necessary improvements, and its use for documentation.

To transcribe the recordings, we used otter.ai [23] to generate initial transcripts, then we listened again to the recordings and corrected mistakes. We conducted a bottom-up thematic analysis by finding codes and themes that emerged from the data [24]. Two of the involved researchers conducted a data analysis session to derive the final themes.

E. Threats to Validity

Internal. A potential threat to internal validity is the researcher's bias in the qualitative analysis of the think-aloud data. To mitigate that, we had one researcher systematically analyze the transcripts of the recordings and extract codes and themes. Another researcher was involved in a data analysis session to check the analysis and derive extracted themes.

Another threat is the transcription accuracy, which affects the data interpretation. To mitigate that, we conducted a manual transcript verification and cleaning process by listening to the original recordings against the transcripts. We corrected transcription errors and the wrong assignment of speaker roles.

Another threat is recording the think-aloud sessions and using an automatic transcription tool. We informed the participants about recording the sessions, their right to withdraw, and how their data would be processed. We received informed consent from all participants. The videos were stored locally, and access to them was limited to only the two researchers who analyzed the data. We only used the automatic transcription tool to transcribe the audio. We selected a tool that is System and Organization Controls 2 (SOC 2) type 2 certified, and adheres to the European General Data Protection Regulation (GDPR), ensuring data security and privacy.

External. A major threat to the generalizability of our findings is the potential influence of external participant selection on the results. Evaluating our approach required a specific background and expertise in cyber-physical systems, which necessitated reaching out to our network. We ensured the selection of participants from multiple companies in different domains. It allowed us to collect and analyze different perspectives on the approach.

IV. Supporting Risk Assessments (RQ1)

A. Requirements for a Risk Assessment Approach

The requirements were derived from the discussed challenges during the workshop and in follow-up meetings in cycle 1.

Req 1: The approach shall support early risk assessment with subsequent iterations: Research recommends using risk assessments at the early-design stages of projects and updating them iteratively; however, risk assessments were done later in most projects. Company A often uses failure mode and effects analysis (FMEA). FMEA was based on historical data and experience from previous projects. There is a lack of understanding of the components involved in robotics missions.

Req 2: The approach shall have a minimal dependence on previous data and experience: With the growth of industry 5.0 [25] and the introduction of new intelligent and collaborative robotics systems, it is challenging to have previous data or expertise that safety analysts can rely on. The approach should be applicable to greenfield development.

Req 3: The approach shall support developers in understanding relevant safety concerns when implementing a mission: It was emphasized that ensuring that the output of risk assessment is adequately considered when implementing robotics missions is challenging and time-consuming. Therefore, our approach should facilitate the tracking of risk assessment outputs and their reflection in the implementation code of robotic missions.

Req 4: The approach shall support visualization: Current approaches make it challenging to obtain an overview of the robotics missions' components and understand the main flow of the missions while doing risk assessments. There is a need to provide visualization and a better understanding of the mission behavior [26].

Req 5: The approach shall support lightweight documentation in an easily accessible way: Previous research has found that ensuring traceability and collaboration becomes difficult when multiple organizational units and multiple tools are involved [27]. It would be undesirable if risk assessment findings were stored in a separate document repository, which might lead to information loss when implementing (coding) the missions. Instead, the approach should bridge the gap between developers and risk assessment experts by making risk assessment documentation easily accessible.

B. The Approach of Using Behavior Trees in Risk Assessment

Our approach integrates behavior trees into the risk assessment approach. The reason to use behavior trees is twofold: By using behavior trees in the process, we have two levels of information needed in risk assessments: the actions involved in a mission (execution nodes) and the decision structure showing how actions lead to specific outcomes (control nodes). Having this high level of abstraction of robotics missions with the graphical representation of the decision-making process might facilitate better identification of actions' requirements and possible failures. Behavior trees might also be used in the implementation and execution phases of robotics projects. Investing time in the design phase to conduct risk assessments using behavior trees could be highly effective, feeding the outputs into the subsequent phases of the project.

Figure 2 presents our proposed approach for model-based risk assessment using behavior-tree models. To explain the approach, we use the collaborative robotic mission (see Sect. II) to walk the reader through the steps. To start, the approach relies on creating a preliminary behavior-tree model of the desired robotic mission. In our case, the robotic mission is a collaborative system that prepares a full kit for an operator using the gantry and the robotic arm in a safe and timely manner.

Figure 3 shows our initial behavior-tree model for the mission. The robot has two tasks: attaching the right gripper (`Attach` sequence) and picking and placing items (`Pick&Place` sequence). The robot starts by checking if the right gripper is attached (`GripperNotAttached?` condition) and executes the `Attach` subtree if needed. Otherwise, the robot moves to the `Pick&Place` sequence, where it moves the arm above the desired blue box position (`MoveToAboveA`), scans and identifies the desired item (`Scan`), picks the item (`Pick`), then moves to the autonomous-robot station (`MoveToB`) and places the item (`Place`).

The next step is to choose the preferred template for risk assessment. We relied on a process-FMEA template typically used at Company A. The risk assessment starts by taking each action in the behavior-tree model, like `Pick`, and identifying prerequirements, potential failures, the effects of a failure, its causes, control prevention, control detection, and recommended actions to mitigate the failure. When applying FMEA, assigning severity, occurrence, and detection numbers to calculate RPN can be postponed. It is often challenging to assign the numbers at the early stages of the project without real data from studies to feed the numbers, which is a known shortcoming of FMEA [28].

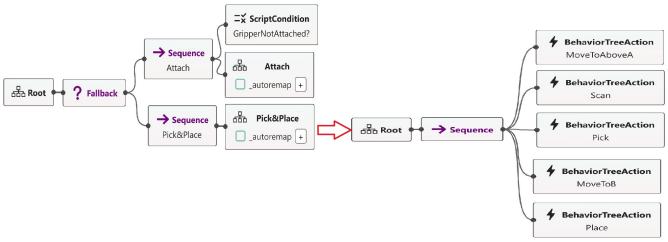

Fig. 3: The BT model representing part of the mission using Groot. On the right is the expanded subtree for Pick&Place.

Table II shows an excerpt of the risk assessments' outputs when applying it to the `Pick` action performed by Team 1. Two possible failures were identified: (1) the desired item is not picked, or (2) the item is dropped after picking. In this case, there are similarities in the characteristics of the failures and future controls.

After having the risk assessment's outputs, the next step is to map them in the behavior trees. In general, most behavior-tree tools offer the possibility to define when nodes fail and succeed, and what to do in case of failure or success. We recommend using the tool Groot, which allows setting the success and failure of a node and provides a "description" field. Using the mapping in Table II, the outputs of FMEA can be added to the behavior-tree model (see Fig. 3 for the model).

Listing 1: An excerpt of `BehaviorTree.CPP` XML notation for Pick action after mapping the risk assessment outputs into behavior trees.

```
<BehaviorTreeAction name="Pick"
action_name="bt_action_service"
command="pick"
_description="Potential effect of failure is
process delay;
potential causes of failure are grasping points not
accurate
or gripper performance deteriorated;
Remember to enable the force/torque sensor to
  detect if the item is in the gripper"
_successIf="ItemPicked"
_failureIf="NoItemPicked: to detect enable
the force/torque sensor if the item is
  in the gripper;
```

Listing 1 shows an excerpt of `BehaviorTree.CPP` XML-like language that is auto-generated by Groot after saving the behavior-tree model. The XML-like file is usually imported as an external file in the implementation code to provide the syntax tree. Mapping the failures into the behavior-tree tool is straightforward. The failures can be mapped into the "_failureIf" field with the control detection information. The remaining characteristics of failures can be mapped into the "_description" field. The recommended mitigation in case of failure is mapped into the "_onFailure" field.

Finally, the final step is providing the mapped outputs in behavior trees to the developer to implement (code) the action nodes. Having the potential risks and related information should guide the developers during implementation. Risk assessment

TABLE II: An excerpt of FMEA results. The mapping field shows the template for transferring FMEA outputs into the behavior-tree tool.

	FMEA	example	mapping
	part	pick	
	pre-requirement	blue box is scanned; the right gripper is attached; an order for picking items is sent by the operator/system	"_description"
characteristics of failures	potential failure mode potential effect(s) of failure potential causes of failure	(1) item not picked; (2) item dropped (1 and 2) process delay (performance affected) (1 and 2) grasping point not accurate or gripper performance deteriorated;	"_failureIf" "_description" "_description"
future controls	controls detection controls prevention recommended action	(1 and 2) to detect, enable the force/torque sensor to check if the item in the gripper, or operator detects failure; (1 and 2) NA (1 and 2) after # of attempts stop execution and notify operator	"_failureIf" "_description" "_onFailure"

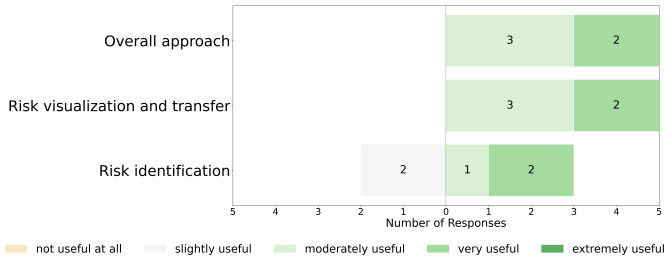

Fig. 4: Responses on the usefulness of our approach.

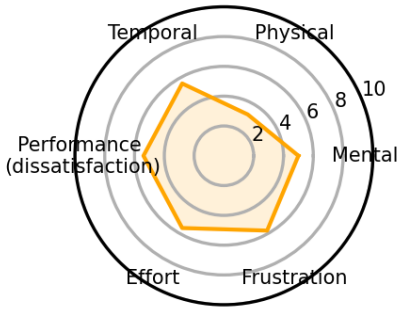

Fig. 5: Perceived workload for using our approach.

is an iterative process. Other risks may arise during the project implementation and evaluation phases, so we recommend keeping the behavior-tree model updated.

V. Evaluation Results (RQ2)

Figure 4 presents the results about the usefulness of our approach reported by the external participants in cycle 2. It can be seen that the participants found the overall approach of using behavior-tree models in risk assessment very useful. The usefulness was considered higher for the overall approach and for the risk visualization and transfer than for the risk identification.

Figure 5 shows an overview of the perceived workload by the external participants after using the behavior-tree-based approach for risk assessment. The dimensions are ranked according to the dimensions of NASA Raw TLX [22], where higher numbers correspond to a higher workload. Overall, the participants experienced a moderate level of perceived workload (average 5.1 on a 1–10 scale), and their experiences were consistent (standard deviation 1.6).

Mental Model for Risk Identification. All participants found it helpful to have the behavior-tree model as a starting point to familiarize themselves with the robotic mission. The behavior tree provided an entry point for thinking about what can go wrong in the robotic mission. It was reported that organizing actions and decisions in a tree-like manner made the potential failure points clearer.

While identifying risks during the FMEA, we noticed two patterns in participants. P1 and P2, who had previous experience with the system RITA, did not use the behavior-tree model as intensely and found the provided model to be somewhat simplistic. At the same time, one participant with prior experience with RITA (P3) and two participants with no prior experience with the system (P4 and P5) appreciated having the model while eliciting the needed information for FMEA. It was stated that the behavior-tree model helped them build their mental model of the mission steps:

P4: I can go back to the figure when I am stuck, because sometimes I can infer what can go wrong. [...] You need the visual sometimes to just remember things. [...] It's like any supporting artifact, even if you know it, sometimes you need some kind of confirmation, so you go back and try to read it one more time while you are thinking.

We found it essential to remind stakeholders using the behavior tree that it is a living model that should evolve as risks are identified, e.g., by adding components to mitigate the risks. For P3 and P4, having the behavior tree might have restricted their brainstorming, and we reminded them that they could add things and that the model was an initial one.

> Finding 1. Using behavior trees at the early stages of risk assessment enables the representation of both the system's intended nominal behavior and potential failure points within a single, structured model, which can help practitioners create a mental model of the mission and associated risks.

Documentation and Version Control. All participants stated that they saw benefits in using behavior-tree models in the risk assessment approach for documentation, even if it is not ultimately implemented as executable behavior trees in the project. However, P5 expressed concerns regarding tracking changes and keeping information up to date across different locations, especially in a regulated industry.

During the think-aloud sessions, P3, P4, and P5 did not

initially transfer all the information from the behavior tree. This shows the possibility of information loss and the need for automation. P4 and P5 expressed the need to automate the mapping of risk assessment outputs into the behavior-tree tool.

P5: Now you're duplicating information that's going to be in multiple places. [...] If I were in a completely unregulated field, that's not really a big deal. [...] As I move into more complex systems and regulated industries, this can be a challenge if you don't have automation in place. [...] From a long-term maintenance perspective, it becomes a challenge in the regulated industries.

Our iterative approach suggests updating and remodeling the behavior trees as failures are identified throughout the project. P3 expressed the need for a version control system to track changes since the approach is iterative. It was expressed that each behavior tree and its associated FMEA evolve in parallel with each iteration of addressing identified failures, necessitating systematic tracking of changes to maintain consistency and traceability.

> FINDING 2. Behavior-tree models are promising for holding and documenting risk assessment information. However, to maintain consistency and traceability, information transfer to existing tools needs to be automated, and a version control system needs to be established.

Granularity of the Model. In the think-aloud tasks, we used a low level of granularity (fewer details) to provide a clear and easy-to-follow behavior tree. P1, P2, P3, and P4 desired the behavior-tree model to have a higher level of granularity (more details) to reduce the need for assumptions.

P4: I was missing some more information to not make many assumptions. [...] I feel it would have been easier with a more comprehensive view of the system behavior.

P4 also noted that with a higher level of granularity, understanding the model might demand extra time, and highly complex and larger behavior-tree models might cause confusion. We acknowledge that it is hard to find the balance between high- and low-granularity models to provide clear robotics missions without being simplistic. Additionally, providing a detailed behavior-tree model can be challenging with new systems and may skew the brainstorming of potential failures. More research is needed to determine the balance in the granularity of the provided model.

> FINDING 3. The level of granularity of behavior-tree models is challenging to balance to provide a clear and understandable mission without biasing the risk assessment brainstorming.

Tooling for Behavior Trees. There were ambivalent experiences with the used behavior-tree tool (Groot). Groot is widely used for behavior-tree modeling, and all participants appreciated the GUI and visualization. However, the tool was not appreciated for transferring and storing the identified failure information. The tool is not designed specifically to hold risk assessment information, which may have hindered the user experience.

P4: I just found the tool to not be user-friendly, and that slightly affected my overall perception of using the BTs for these tasks negatively.

In general, behavior trees are appropriate models to identify risks and potential failures, as they require defining when nodes succeed or fail. Thus, most existing tools will have fields to map this information. We found Groot to be beneficial, partially because of the available data fields for the identified failure risks and recommended actions. One challenging aspect was mapping the rest of the outputs (prerequirements, effect and cause of failure, and controls detection and prevention). Since Groot had a free-text field for description for each node, we mapped the information into it. However, it is not necessarily an optimal way to present the information. P1, P3, and P4 highlighted that connecting information recorded in the free-text field with that entered into structured fields for specific failures might lead to traceability issues, as all this information pertains to the same failure event.

P3: It's a bit hard when you have this free text... I need some way to make it clear which failure is related to the information in the description.

All participants expressed the need for better tooling that connects risk assessment artifacts to code in a smoother way. P2, P3, and P4 stated that it would be beneficial to conduct risk assessment directly in the behavior-tree tool—partially because it would save time and cognitive effort, and because it would make it easier to connect the identified information in the risk assessment directly to the code. All participants found switching between tabs problematic. When the outputs of the risk assessments were transferred into the behavior-tree tool, participants reported losing the overview of information compared to the Excel worksheet. One participant preferred entering all information related to a single failure at once.

P3: Maybe you could add an FMEA tab here, and then you have all this, exactly these headlines.. then you can say that you follow the FMEA... an added value here is the connection to the source code.

We believe the aforementioned positive aspects and limitations expressed could be used as requirements to extend existing behavior-tree tools to support the early identification of risk information that influences the code.

> FINDING 4. The behavior tree tool Groot could naturally model information about potential failures and recommended actions from FMEA, but it lacks optimal support for other types of information. Additionally, the choice of the behavior-tree tool impacts the process of transferring and visualizing the outputs of risk assessment. This highlights an opportunity for the modeling community to improve existing tools.

The Gap Between Risk Assessment and Implementation. In the internal evaluation in cycle 1, the developer used the enriched behavior-tree model in Groot to implement (code) the robotic mission. The enriched behavior-tree model conveyed nominal behaviors, expected sequences, possible failure modes,

and contingency actions for various scenarios—all of which stem directly from the risk assessment. This approach ensured that the developer had direct access to comprehensive information about both standard and failure scenarios, removing ambiguity and reducing the need for interpretative guesswork.

The external participants in cycle 2 did not perform the task of implementing the resulting behavior trees. However, P5 anticipated the potential of having the enriched behavior-tree model with risk assessment information to improve the development process—a perspective substantiated by the developer's practical experience. In addition, P2 and P3 noted that they valued the ability to connect and transfer risk information to the code for later stages in the projects.

P5: It always comes down to how experienced your engineers and team are. Having the right assumptions or mental model going in. When I hire new college grads or software engineers, I can't expect them to know all the failure modes. So, it can be a lot of a higher value to have the behavior tree with risks right in front of them.

> FINDING 5. We can ensure alignment between high-level risk assessments and low-level implementation (code) by leveraging behavior-tree models as an information bridge for risk assessment outputs.

VI. CONCLUSION AND FUTURE WORK

This work is the first step to support risk assessments using behavior-tree models. We developed and provided a model-based approach for using behavior trees in risk assessments with the support of industrial practitioners. Our approach uses the behavior-tree model to support risk identification, visualization, and transfer of risk assessments' outputs. We evaluated the approach with practitioners from different companies. Our findings highlighted the potential of the behavior-tree model in supporting practitioners in forming a better understanding of the robotic mission, thereby identifying potential failure points in the early stages of the project. They also identified the potential of aligning code implementation with the risk assessment outputs and forming lightweight documentation. However, further development of our approach is needed to automate the transfer of risk assessment information to implementation, select the appropriate model granularity, and provide enhanced tooling.

In future work, we want to measure the improvement in identifying risks when using behavior trees compared to traditional risk assessment processes. Finally, we want to improve existing tooling to support information transfer.

ACKNOWLEDGMENT

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] S. García, D. Strüber, D. Brugali, T. Berger, and P. Pelliccione, "Robotics software engineering: A perspective from the service robotics domain," in *ESEC/FSE*, 2020.

[2] J. Guiochet, M. Machin, and H. Waeselynck, "Safety-critical advanced robots: A survey," *Robotics and Autonomous Systems*, vol. 94, 2017.

[3] E. Tom, A. Aurum, and R. Vidgen, "An exploration of technical debt," *Journal of Systems and Software*, vol. 86, no. 6, 2013.

[4] H. Foidl, M. Felderer, and S. Biffl, "Technical debt in data-intensive software systems," in *SEAA*. IEEE, 2019.

[5] P. Curtis, M. Carey, C. of Sponsoring Organizations of the Treadway Commission *et al.*, "Risk assessment in practice," 2012.

[6] A. Marzinotto, M. Colledanchise, C. Smith, and P. Ögren, "Towards a unified behavior trees framework for robot control," in *ICRA*, 2014.

[7] M. Colledanchise and P. Ögren, *Behavior trees in robotics and AI: An introduction*. CRC Press, 2018.

[8] "Online appendix," 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15630722

[9] R. Ghzouli, T. Berger, E. B. Johnsen, A. Wasowski, and S. Dragule, "Behavior trees and state machines in robotics applications," *IEEE Transactions on Software Engineering*, vol. 49, no. 9, 2023.

[10] *Directive 2006/42/EC of the European Parliament and of the Council on Machinery, and Amending Directive 95/16/EC (recast)*, European Parliament and Council of the European Union, Brussels, May 2006, official Journal of the European Union.

[11] *Safety of machinery – General principles for design – Risk assessment and risk reduction*, International Organization for Standardization, Geneva, Switzerland, November 2010.

[12] M. Bdiwi, I. Al Naser, J. Halim, S. Bauer, P. Eichler, and S. Ihlenfeldt, "Towards safety4. 0: A novel approach for flexible human-robot-interaction based on safety-related dynamic finite-state machine with multilayer operation modes," *Frontiers in Robotics and AI*, vol. 9, 2022.

[13] N. Banduka, I. Veza, and B. Bilić, "An integrated lean approach to process failure mode and effect analysis PFMEA: A case study from automotive industry," vol. 11, no. 4, 2016.

[14] H. Schneider, "Failure mode and effect analysis: Fmea from theory to execution," 1996.

[15] D. Battini, M. Calzavara, A. Persona, and F. Sgarbossa, "A comparative analysis of different paperless picking systems," *Industrial Management & Data Systems*, vol. 115, no. 3, 2015.

[16] A. Abdulkhaleq and S. Wagner, "Integrating state machine analysis with system-theoretic process analysis," in *Software Engineering 2013-Workshopband*. Gesellschaft für Informatik eV, 2013.

[17] G. Kokotinis, G. Michalos, Z. Arkouli, and S. Makris, "A behavior trees-based architecture towards operation planning in hybrid manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 37, no. 3, 2024.

[18] L. Castano and H. Xu, "Safe decision making for risk mitigation of uas," in *ICUAS*. IEEE, 2019.

[19] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, 2004.

[20] K. Säfsten and M. Gustavsson, "Research methodology: for engineers and other problem-solvers," 2020.

[21] F. Shull, J. Singer, and D. I. Sjøberg, *Guide to advanced empirical software engineering*. Springer, 2008, vol. 93.

[22] S. G. Hart, "Nasa-task load index NASA-TLX; 20 years later," in *Proc. of the human factors and ergonomics society annual meeting*, vol. 50, no. 9. Sage publications Sage CA: Los Angeles, CA, 2006.

[23] Otter.ai, "Otter.ai: Transcription service," accessed April 23, 2025.

[24] D. S. Cruzes and T. Dybå, "Recommended steps for thematic synthesis in software engineering," in *ESEM*. IEEE, 2011.

[25] V. Toncian, A. Florea, A. David, D. Morariu, and R. Cretulescu, "Leveraging collaboration for industry 5.0: Needs, strategies and future directions," in *Working Conference on Virtual Enterprises*. Springer, 2024.

[26] J. Hutchinson, J. Whittle, and M. Rouncefield, "Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure," *Science of Computer Programming*, vol. 89, 2014.

[27] R. Wohlrab, E. Knauss, J.-P. Steghöfer, S. Maro, A. Anjorin, and P. Pelliccione, "Collaborative traceability management: a multiple case study from the perspectives of organization, process, and culture," *Requirements Engineering*, vol. 25, no. 1, 2020.

[28] H.-C. Liu, L. Liu, and N. Liu, "Risk evaluation approaches in failure mode and effects analysis: A literature review," *Expert systems with applications*, vol. 40, no. 2, 2013.