

Article

Execution Plan Control in Dynamic Coalition of Robots with Smart Contracts and Blockchain

Nikolay Teslya ^{*,†}  and Semyon Potryasaev [†] 

SPIIRAS—St. Petersburg Institute of Informatics and Automation, Russian Academy of Sciences, 14th line, 39, 199178 St. Petersburg, Russia; spotryasaev@gmail.com

* Correspondence: teslya@iias.spb.su

† These authors contributed equally to this work.

Received: 30 November 2019; Accepted: 31 December 2019; Published: 4 January 2020

Abstract: The paper presents an approach of the blockchain and smart contracts utilization for dynamic robot coalition creation. The coalition is forming for solving complex tasks in industry applications that requires sequential united actions from the several robots. The main idea is that the process is split into two stages: scheduling and dynamic execution. On the scheduling stage, the coalition is defined based on the correlation of existing tasks and robot equipment, and the execution plan is formed and stored in smart contracts. The second stage is the plan execution. During this stage, smart contract controls how each robot solves its sub-task and whether it solves the sub-task due to the planned moment of time. In case of any deviation from the plan, smart contacts will provide a solution for returning to the plan or for changing the coalition composition with new robots and an execution plan. The prototype for execution control system has been developed based on the Hyperledger Fabric platform.

Keywords: dynamic coalition; robot; blockchain; smart contract; execution control

1. Introduction

Nowadays, robots have become complex enough to cooperate for solving complex tasks [1]. Starting from the simplest ways of cooperation like swarms or flocks inspired by the nature, current development had moved to a more complex socio-inspired type of cooperation: coalition task solving. Unlike simple swarm and flock models, where robots are identical in structure and software, a coalition of robots can be made up of devices with different functionality. In addition, each robot is unique in terms of control algorithms, making a decision based on the analysis of the current situation. The independence of the robots in the coalition allows them to have their own goals and rely on them. The solution to each complex task is complemented by a list of required resources, the number of which is limited, and a reward for solving it. When jointly solving tasks, robots need to coordinate with others in order to gain access to limited resources and maximize their gain.

Manufacturing on demand, precision agriculture, and remote exploring (including aerospace scan of planets) can be considered as examples of the areas of possible use of robot coalitions that require the coordinated action of robots with different functionalities [2–4]. For example, in precision farming, there are robots capable of sowing, watering, fertilizing, exploring a territory, analyzing the composition of the soil that should work together to get maximal benefit. The benefit in precision farming is to obtain the maximum yield when growing different crops, spending a minimum of resources. For this, it is necessary to determine the best conditions for each crop and plan the sequence of actions for the care and harvest [5].

One of the specific features of the coalition is the accessibility of complete information to all its participants. They have access to information about the current state of the coalition, the status of solving problems, available and used resources, and the distribution of reward. A prerequisite is the

ability to quickly familiarize the robot with the current state of problem solving and tracking the procedure for performing actions. At the same time, it is necessary to ensure a consensus among all coalition members on the order of execution of actions and the immutability of the problem solution history for subsequent analysis.

Interaction between coalition participants requires the transparency of operations related to the production, supply, receipt, and processing of the task for execution plan control [6]. In this paper, the use of blockchain technology is proposed to organize a decentralized distributed immutable storage where the actions log will be stored. Expanding the possibilities of the ledger through the use of smart contracts makes it possible to store specific rules of action in various situations as well as the rules of interaction between participants that will also be available within a coalition.

The paper contributes in the field of coalition task solving by describing the plan control approach based on utilizing blockchain and smart contract technologies. The two-step process is described for task solving: coalition formation with task scheduling and plan execution. The plan is stored in the blockchain as smart contracts that control the success of plan stages. In case of plan failure, smart contracts are used to adjust execution or reschedule the stages.

The rest of the paper is structured as follows: Section 2 provides the problem statement with an overview of existing solutions. Section 3 contains the main principles of coalition formation and scheduling for robots to solve the common task. Section 4 describes how blockchain and smart contracts are utilized to control plan execution. Sections 5 and 6 provide examples and discussion of the proposed approach and conclusions over the example. Sections 7 and 8 provide materials used to acquire the results in the paper and conclusion over the approach presented in the paper.

2. Problem Statement

The task set in this paper is to provide a joint task solving in dynamically changing environments by robots in a coalition. The process should be controlled by all coalition participants to track changes in coalition and environment state; therefore, it is necessary to provide common storage of the current coalition state, task solving state, as well as resource and reward distribution.

The coalition task solving is viewed as a two-stage process: scheduling and execution. In the first stage, the coalition is formed and the execution plan is built in a static or dynamic way by analyzing the task requirements, robot competences, and environment characteristics. To form the coalition, at least two robots are required. The plan contains a list of sub-tasks for each robots with execution order and timestamps when each sub-task should be solved. Since some sub-tasks should be solved consequentially, all robots have to track the current status of task solving and coordinate their own action with the others. After the coalition was formed, the execution stage is starting. During this stage, all robots perform their actions and report to the others about the progress on each task.

Interaction of robots during the plan execution should be carried out through a common information space synchronized between all of the coalition members. Records of the completion of each plan phase should be maintained in this space so that they can be verified and used to monitor the plan execution. The information space should also contain the current amount of resources consumed by the coalition in the course of the task and their distribution among the coalition members.

One of the requirements of the problem is the ability to automatically adjust the plan when a deviation is detected. It should be noted that the adjustment may be partial—when the next stages of the plan are redistributed among the coalition members in such a way as to smooth out the deviation, or complete, as a result of which a new coalition is formed, for which a new plan for completing the subtasks is defined. Information on the implementation of the plan should be kept unchanged in the information space for the subsequent analysis of the causes of deviation, while the impossibility of their change should be guaranteed.

3. Task Scheduling for Coalition

The section will describe basic concepts for dynamic robot coalition formation and task scheduling. First, the coalition formation is shown. When the task appears, one of two possible scheduling process is starting: static or dynamic.

3.1. Coalition Formation

Coalition is the most highly organized type of collective work organization [7]. Each robot in coalition is considered as an independent agent with its own competencies and goals, which he aims to achieve after the problem solving. The coalition in this case can be considered as a union of agents with their own interests, which in the negotiation process make a decision on a joint solution of the problem and the distribution of the reward.

When forming a coalition, it is necessary to take into account the capabilities of robots in such a way that in total they cover all the requirements of the task to be jointly solved. To this end, it is proposed to use an ontology that provides descriptions of the requirements of tasks and the capabilities of robots [7]. An example of an ontology for describing a robot is presented in Figure 1. The presented ontology is adjusted to the specific task of obstacle overcoming by a coalition of robots, but the main concepts can be adjusted with upper level ontology for robotics and automation (IEEE CORA [8]) in order to provide higher interoperability with other ontologies.

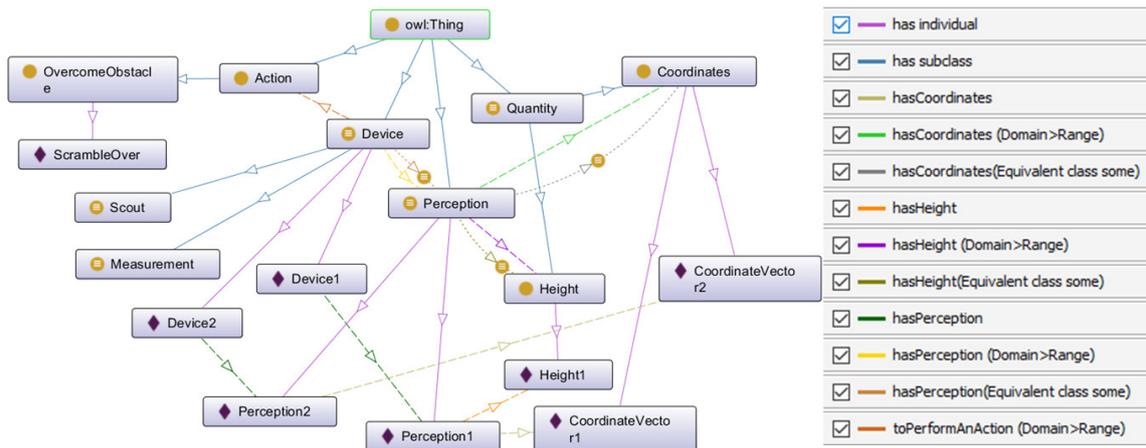


Figure 1. Example of robot ontology.

In order to select robots for the task, the reasoning based on the SWRL (A Semantic Web Rule Language [9]) rules is used. An example of using the rules is presented below in Section 4. An additional criterion for the selection of robots can also be an expectation in the distribution of reward across coalition members, for example by using fuzzy cooperative games approach [10,11], but this issue is beyond the scope of this work. After the formation of a set of robots capable of jointly solving the task, implementation scheduling is carried out, presented in Section 3.2. The overall process of joint task solving in a coalition is shown in Figure 2.

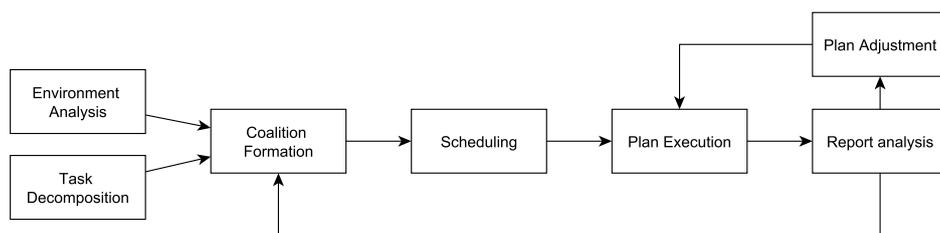


Figure 2. Scheme of joint task solving.

3.2. Scheduling

Three approaches for scheduling have emerged by now [12]: satisfactory (incremental), formal, and system (comprehensive) planning. Various planning theories realizing the above-mentioned concepts for different applications have been developed by now [12–14].

Here, the main focus will be given to incremental and formal process planning. It is proposed to use the Business Process Model and Notation (BPMN) to create a schedule of joint task solving. The BPMN is mostly used for the description of business-processes and is familiar to both technical specialists and business users. It also can be used in the context of integrated modeling of task execution processes. Among the advantages of BPMN, the following ones are important for complex task scheduling in the proposed approach: the set of the applied primitives in BPMN combines the advantages of other notations and allows for representing the models of distributed processes; it provides a wide range of capabilities for formal representation of components of complex processes [15].

The processes described in BPMN can be used to carry out both analytical and simulation modeling with application of corresponding software environments. The basis of BPMN provides not enough declared attributes to carry out analytical modeling. However, due to extensibility of the BPMN, additional attributes important for the incremental and formal process planning, like estimation start or end time for the task, can be added without losing backward compatibility with its runtime environment. Recorded in BPMN complex process with all necessary additional attributes can be executed in the earlier-developed application of analytical modeling based on dynamic interpretation of the processes of carrying out the operations and distributing the resources of complex objects.

The advantage of an extension of the BPMN application area consists of considerable reduction of labor intensity of basic data input while conducting analytical modeling of existing processes in complex task solving. While speaking on specified advantages, it is necessary to mention, firstly, that BPMN is focused on the simplification of data input and their visualization due to the availability of graphic representation and a limited number of concepts. Secondly, many enterprises already have processes described in this notation that simplify schedule development; and, consequently, preparation of basic data for analytical modeling is limited to an introduction of some additional concept attributes. Thirdly, the area of automatic creation of diagrams described in the form of a text form is being developed [16]. For example, a number of works informing on successful implementation of the method of creation of BPMN based on a sequence of actions description in the form of text are known [17,18]. In addition, there appears a possibility to apply modern technologies of modeling automation at all stages of complex modeling implementation.

Taking into account the described features of BPMN, it is proposed to use it for static simulation of solving complex problems in a coalition of robots. A model built using BPMN will reflect the main stages of solving a complex task by dividing it into simple subtasks, which will allow robots to be selected for each of the subtasks in accordance with its description and equipment requirements and resources for the solution. An example of a production process model in BPMN is presented in Figure 3.

To fill the BPMN model with timestamps of the plan stages start and end, dynamic operation planning should be used. The general technology of dynamic operational planning includes the following stages: structural and parametric adaptation of planning models, and algorithms to the past and present states of coalition and to the past and present states of the environment; structure-functional synthesis of the main coalition elements; scheduling, simulation of possible scenarios of coalition functioning according to the schedule, structural and parametric adaptation of the schedule, models, and algorithms to future coalition states and environment states (predicted via simulation models). The operation planning is based on the previous works [19,20].

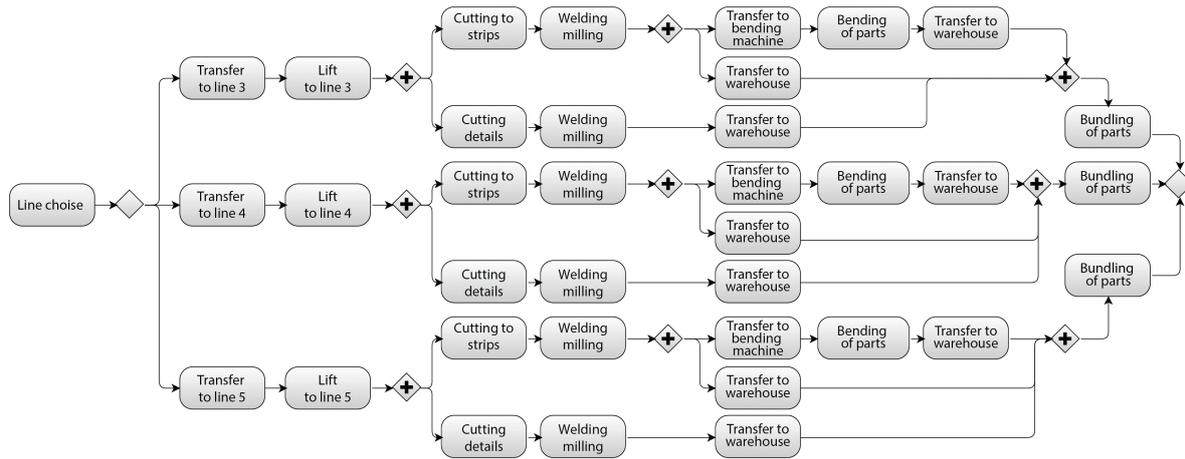


Figure 3. Fragment of the manufacturing process in BPMN.

4. Plan Execution Control

A diagram of the proposed information space with support for blockchain and smart contracts is presented in Figure 4 [21]. In this scheme, there are two parts—physical and cyber environments. The physical environment represents the components of the coalition and the parts of the environment, which are physical objects, for example, robots, their components, physical resources, environmental parameters, like temperature, pressure, humidity, etc. In the cyber environment, information components are presented that are associated with control programs represented by intelligent agents (Knowledge processors) united by smart space based on a “blackboard” concept. This approach allows for combining robots with different characteristics from different manufacturers, described using ontology.

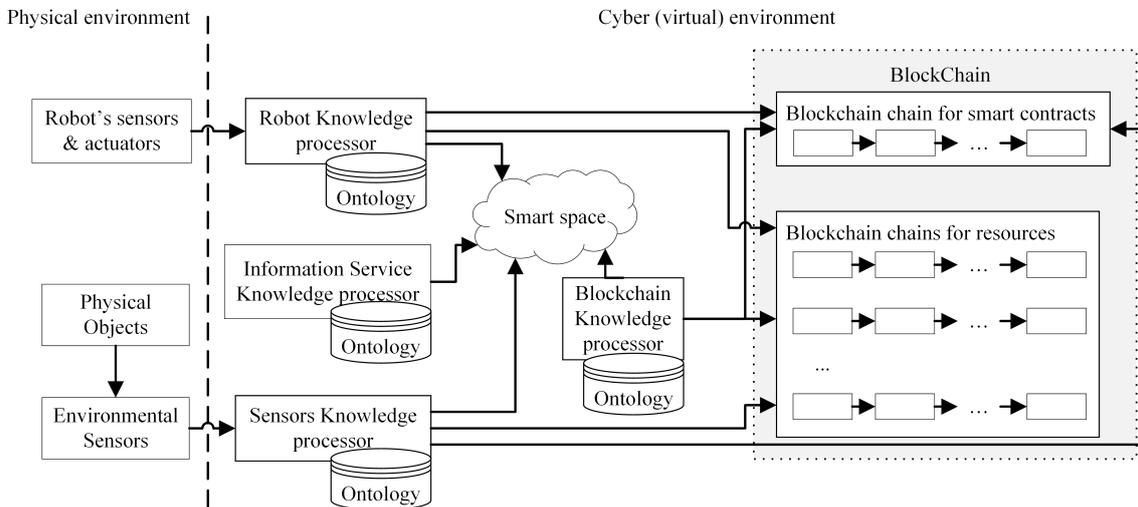


Figure 4. Framework for robot coalition interaction.

Coalition formation is carried out on the basis of the use of information from the blackboard and the capabilities of robots and the required functions for solving the problem using SWRL rules. For example, the following rule (see Listing 1) is used to select robots located along the $(x1, y1)$ and $(x2, y2)$ coordinates, one of which must be a quadcopter and the other a rover that is capable of scanning the area and overcoming obstacles.

Listing 1. Example of SWRL rules to select a quadcopter and a rover with defined requirements.

```

Device(?d2) ∧ Device(?d1) ∧ hasCoordinateX(?v1, ?vx1) ∧ hasCoordinateX(?v2, ?vx2) ∧
hasCoordinateY(?v1, ?vy1) ∧ hasCoordinateY(?v2, ?vy2) ∧ Perception(?p1) ∧
isQuadcopter(?d1, ?true) ∧ swrlb : equal(?vx1, ?vx2) ∧ swrlb : equal(?vy1, ?vy2) ∧
isRover(?d2, true) ∧ hasHeight(?p1, ?h1) ∧ hasPerception(?d1, ?p1) ∧ OvercomeObstacle(?a) ∧
hasCoordinates(?p1, ?v1) ∧ hasCoordinates(?p2, ?v2) → toPerformAnAction(?d2, ?a)

```

4.1. Blockchain Usage for Robot Coalition

The existing approach to ensuring the interoperability of components through the cloud-based IoT platform with using ontologies was expanded by adding a distributed digital transaction ledger based on blockchain technology to meet requirements for security, long-eternity, and information unmutability (see Figure 4). Each knowledge processor is also available to interact with other components through the use of the blockchain, which allows for tracking the current state of the process in which it participates and updates the information about the part of the process for which the relevant component is responsible.

The role of the blockchain in this approach is to control the allocation of resources necessary for the joint solution of the problem and the storage of the history of actions of coalition members. The architecture and mathematical models behind the blockchain allows for the immutability of records and the rapid distribution of records among all coalition members. This allows minimizing the delays between the stages of solving a complex task and ensuring the completeness of information on the current status of the decision for all coalition members.

In the framework depicted in Figure 4, two groups of chains are distinguished within the blockchain. The first one stores smart contracts, providing access to the history of their changes to all participants. The second group stores tokenized resources [22]. Each resource corresponds to a separate chain in which the entire distribution history is stored. This ensures the timely update of information between all robots and the possibility of subsequent analysis of the process of solving the problem, in order to increase the efficiency of resource use.

In addition to support consumables control in an industrial socio-cyberphysical system, smart contracts can be used for other purposes. For example, they can act as a representative of a certain component.

4.2. Smart Contracts for Execution Control

In platforms that implement blockchain technology, the capabilities of the ledger can be expanded through the use of smart contracts [23]. The smart contract is defined as: “a set of promises, specified in digital form, including protocols within which the parties perform on these promises” [24]. A smart contract within blockchain technology can be viewed as a decentralized application available to all participants of the blockchain. For implementation of additional protocols over an ordered list of published statements, for example, ownership and transfer of consumables, it is necessary to keep a certain state and track its change according to generally accepted rules in accordance with published statements. State changes of coalition are reflected in the smart space by means of adding/deleting relevant information. During the interaction between coalition participants through the IoT platform with the blockchain support, all smart contracts can be called either directly through the provided transaction initiation interface in the blockchain or using other smart contracts methods. To simplify the use of custom protocols, the method of any contract should be called using the basic smart contract. For this purpose, a basic smart contract was developed in the detachment, which provides support for the following functions:

1. Schedule upload using XML format. It provides:
 - (a) Receiving an execution plan in XML.
 - (b) Parse it, extract the robots, tasks associated with robots, the order of execution, and the timing of tasks.
 - (c) Generation entries from the blockchain from the extracted items. Each entry must contain one task, which is described by the robot responsible for its execution and the start and end dates of the execution.
 - (d) Send on the blackboard a notice of the start of execution.
2. Start the task. Accept a message from the robot about the start of the task, record the moment of the real start of execution in the blockchain, verify with the planned one, and record the fact in case of a strong deviation.
3. Completion of the task. Same as with the start—get notification, check correctness and write to the ledger. In case of a strong deviation from the plan, write to the blockchain and notify others about the failure of the plan through the blackboard.

The basic smart contract is available through the REST API so each robot can access the plan without direct connection to the blockchain. In addition, a simple web page is available for a human operator to check the coalition state and follow the process of joint task solving.

4.2.1. Dynamic Execution Adjustment

After receiving a scheduled task, the robot notifies the coalition of the start of its implementation, by calling the main smart contract containing the processed schedule. At the end of the task, the robot returns a signal of completion and the smart contract calls the other robots and environmental sensors to confirm the completion of the task. The confirmation is based on the BFT consensus protocol over the distributed ledger [25]. After confirming the completion of the task, information about the execution time is entered in the registry. Three situations are possible here. The first one is ideal case, when all processes continue as planned (see in Figure 5). In this case, the operation will just continue without any changes. The second one is raised in case of some the robots are late with their response for some reason. If the total time is different from the planned one, but not more than the allowed interval (for example, the process manager can set the maximum deviation to one minute or some heuristic can be used to estimate overall delay of joint problem solving), then the time of the remaining operations is not recalculated and coalition just postpones all scheduled intervals for subtask solving based on the value of delay. Otherwise, the third case is raised and the schedule must be recalculated for the remaining operations.

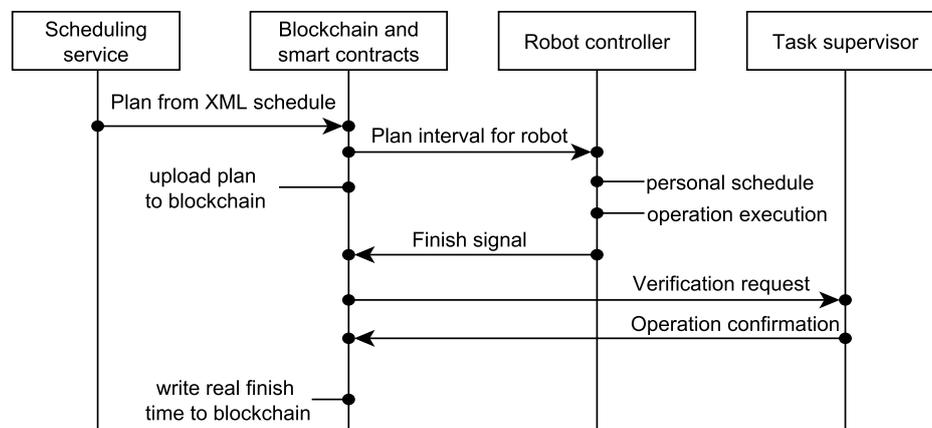


Figure 5. Process for adjustment.

4.2.2. Rescheduling

If the rules enshrined in the contract do not allow for a return to a previously planned schedule, the rescheduling is required. The decision about rescheduling is made based on an analysis of structure-dynamics control in a complex dynamic object that is the coalition in the paper. The control is based on the analysis of how the current coalition status influences maps to the reachability area calculated based on the initial state of coalition and task properties to be solved jointly [26]. To do this, the main smart contract provides for a reassessment of the task execution time each time the status for individual subtasks is updated. Such a situation may arise if the smart contract did not receive confirmation of the completion of the task at the expected time point, taking into account the allowed interval. This may be due, for example, to the physical failure of a robot. In this case, the smart contract sends a notification to complete rescheduling of the process that starts with the formation of a new coalition by replacing a robot with the same characteristics or by a group of robots that satisfy the task requirements and then a new schedule is calculated. A diagram of the action for rescheduling the plan is presented in Figure 6.

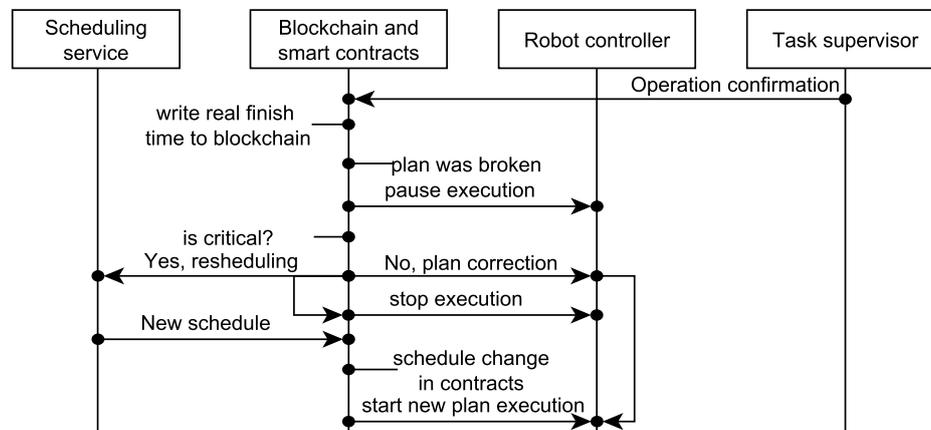


Figure 6. Process for rescheduling.

5. Results

This section presents an example of a schedule file generated from a BPMN diagram (see Listing 2). It reflects the main operations performed to solve a common task. For each operation, the estimated start and end time is shown as well as the required resource for task solving. To solve the problem in a complex manner, the quality metrics are determined, related to the execution time and energy consumption (i.e., minimal energy consumption of whole coalition or minimal execution time [27]).

The resulting file is transferred to the smart contract on the Hyperledger Fabric platform, which is parsed by the main smart contract in the blockchain in the following way:

1. The schedule object is a task. Combines processes, quality indicators, and resources. There are no special properties and identifiers.
2. Array of processes. Each process contains its own identifier and an array of operations.
3. Each operation in the array contains an identifier, name, priority, a link to the involved resource, and several possible moments of the beginning and end of the operation.
4. An array of quality indicators, contains Pki objects. Each such object has an identifier and property generalized, weight, name, and value.
5. Array of resources. Each resource contains an identifier (it is accessed from the operation, see above) and worktime.

Listing 2. Example of BPMN file for the coalition process description.

```

<spy:schedule>
  <spy:processes>
    <spy:Process id="1">
      <spy:operations>
        <spy:Operation id="1">
          <spy:name>Transfer to line #3</spy:name>
          <spy:priority>0.3254</spy:priority>
          <spy:resource>RS_2</spy:resource>
          <spy:start>
            <spy:StartTime intensity="5">12</spy:StartTime>
            <spy:StartTime intensity="3">24</spy:StartTime>
          </spy:start>
          <spy:end>
            <spy:EndTime>15</spy:EndTime>
            <spy:EndTime>25</spy:EndTime>
          </spy:end>
        </spy:Operation>
        ....
      </spy:operations>
    </spy:Process>
    ....
  </spy:processes>
  <spy:quality>
    <spy:Pki id="J1" generalized="False">
      <spy:weight>0.8</spy:weight>
      <spy:name>Energy consumption</spy:name>
      <spy:val>321</spy:val>
    </spy:Pki>
    <spy:Pki id="J2" generalized="False">
      <spy:weight>0.2</spy:weight>
      <spy:name>Time consumption</spy:name>
      <spy:val>31</spy:val>
    </spy:Pki>
  </spy:quality>
  <spy:resources>
    <spy:Resource id="RS_2">
      <spy:worktime>12</spy:worktime>
    </spy:Resource>
    <spy:Resource id="RS_3">
      <spy:worktime>16</spy:worktime>
    </spy:Resource>
  </spy:resources>
</spy:schedule>

```

All information about operations is recorded in contracts that track execution time. They are launched by the Hyperledger platform automatically upon the occurrence of an event (specified time) or by invoking by a robot after a task is finished. The work of the contract ensures the continuation of the plan or its adjustment.

6. Discussion

The result obtained in this paper can be used as the basis for organizing the process of joint problem solving by a coalition of robots. The plan for the problem solving is built on the basis of BPMN notation that allows for moving from a visual description of the process to real steps, using

an analysis of the graph of possible solutions, in which a path is selected that corresponds to a solution that satisfies the conditions of the problem (for example, taking into account energy and resource constraints). The constructed plan contains a distribution of coalition members and time stamps for which individual stages of the plan should be resolved. Transferring the plan to smart contracts will allow all coalition members to have access to the list of tasks and to monitor the implementation of the plan through joint efforts.

This solution in the current form still has a number of disadvantages associated with the limitations of calculating a plan that satisfies the conditions of the problem being solved jointly and using a distributed ledger. The limitation of calculating the plan is due to the difficulty of finding the path in the graph, since the number of coalition participants, the number of stages and possible transition paths between stages can form a graph of a rather complex structure. Finding the shortest path in such a column will take considerable time, which makes it impossible to frequently recalculate the solution plan.

The restriction on the use of a distributed ledger is associated with the computational complexity of the task that needs to be solved for making an entry in the ledger and matching this entry between all coalition members. Many robotic systems, due to size limitations, cannot carry a sufficiently powerful computing device on board, which imposes a significant restriction on their participation in the formation of blocks for a distributed ledger. This problem can be solved using the complex structure of a distributed ledger, for example, as in Hyperledger Fabric, in which weak devices are only data providers, and the whole calculation is performed on more powerful devices. All the shortcomings presented here are considered as a direction for further work.

7. Materials and Methods

The computer codes used to derive the results in this paper can be found as open access on [28]. The repository contains contract code on Go language for Hyperledger Fabric platform and example of project BPMN and JSON files.

8. Conclusions

The paper presents an approach that allows for tracking the process of forming a coalition of robots and solving complex tasks by them. The proposed approach is based on the use of blockchain and smart contracts. This allows for closing several issues at once, traditionally solved by a whole set of architectural solutions, providing a secure repository of tasks, tracking the current status of solving a common task, automatically detecting deviations from the plan, and forming control actions on the coalition structure when deviations from the execution plan are detected.

In the future, it is planned to use dynamic process models for analyzing the attainability of solving a task by a coalition of robots at the appointed time. This will allow for getting away from the fixed threshold, after which a complete rescheduling of the solution of the problem with the formation of a new coalition is required. In addition, the case study implementation for various kinds of rescheduling scenarios is planned for the nearest future and the results will be available in open access through the gitlab project referenced earlier.

Author Contributions: All authors contributed equally to conceptualization, methodology, validation, investigation, and writing—original draft preparation and editing. BPMN model extending and scheduling was prepared by S.P. Coalition formation mechanism, execution control, dynamic execution adjustment, rescheduling was prepared by N.T.

Funding: This research was funded by RFBR Grant No. 17-29-07073 and by Russian State Research No. 0073-2019-0005.

Acknowledgments: The authors want to acknowledge Maxim Shchekotov for assistance in robot ontology development and Oleg Kofnov for his great contribution in smart contract implementation for HyperLedger platform.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Panov, A.I.; Yakovlev, K. Behavior and Path Planning for the Coalition of Cognitive Robots in Smart Relocation Tasks. In *Robot Intelligence Technology and Applications 4*; Kim, J.H., Matson, E.T., Myung, H., Xu, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; Volume 208, pp. 3–20. doi:10.1007/978-3-319-31293-4_1.
2. Balta, E.C.; Jain, K.; Lin, Y.; Tilbury, D.; Barton, K.; Mao, Z.M. Production as a Service: A Centralized Framework for Small Batch Manufacturing. In Proceedings of the 2017 13th IEEE Conference on Automation Science and Engineering (CASE), Xi'an, China, 20–23 August 2017; pp. 382–389. doi:10.1109/COASE.2017.8256133.
3. Shah, A. Robot Explorers. *Mech. Eng.* **2018**, *140*, 30–35.
4. Khanna, A.; Kaur, S. Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. *Comput. Electron. Agric.* **2019**, *157*, 218–231.
5. Roldán, J.J.; del Cerro, J.; Garzón-Ramos, D.; Garcia-Aunon, P.; Garzón, M.; de León, J.; Barrientos, A. Robots in agriculture: State of art and practical experiences. In *Service Robots*; InTech: Rijeka, Croatia, 2018.
6. Ferrer, E.C. The blockchain: A new framework for robotic swarm systems. In Proceedings of the Future Technologies Conference (FTC), Vancouver, BC, Canada, 15–16 November 2018.
7. Kashevnik, A.; Teslya, N. Blockchain-Oriented Coalition Formation by CPS Resources: Ontological Approach and Case Study. *Electronics* **2018**, *7*, 66. doi:10.3390/electronics7050066.
8. IEEE Robotics and Automation Society. *IEEE Standard Ontologies for Robotics and Automation*; IEEE: Piscataway, NJ, USA, 2015; p. 60. doi:10.1109/IEEESTD.2015.7084073.
9. Horrocks, I.; Patel-Schneider, P.F.; Boley, H.; Tabet, S.; Grosz, B.; Dean, M. SWRL: A Semantic Web Rule Language Combining OWL and RuleML <https://www.w3.org/Submission/2004/SUBM-SWRL-20040521/> (accessed on 4 January 2020).
10. Hosam, H.; Khaldoun, Z. Planning coalition formation under uncertainty: Auction approach. In Proceedings of the 2006 International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2006), Damascus, Syria, 23–28 April 2006; Volume 2, pp. 3013–3017. doi:10.1109/ICTTA.2006.1684896.
11. Sheremetov, L.B.; Smirnov, A.V. Supply Chain Configuration as a Cooperative Game with Fuzzy Coalitions. In *Supply Chain Management Under Fuzziness*; Kahraman, C., Öztaysi, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 313, pp. 293–314. doi:10.1007/978-3-642-53939-8_13.
12. Ackoff, R.L. *The Art of Problem Solving*; Wiley: New York, NY, USA, 1998.
13. Klir, G.J. *Architecture of Systems Problem Solving*; Springer: Berlin/Heidelberg, Germany, 2013.
14. Hanziverov, F.; Ostrouhov, V. *Simulation of Space Systems for Natural Resources Research*; Mashinostroenie: Moscow, Russia, 1989.
15. Pereira, J.L.; Freitas, A.P. Simulation of BPMN process models: Current BPM tools capabilities. In *New Advances in Information Systems and Technologies*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 557–566.
16. Deeptimahanti, D.K.; Sanyal, R. An innovative approach for generating static UML models from natural language requirements. In Proceedings of the International Conference on Advanced Software Engineering and Its Applications, Sanya, China, 13–15 December 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 147–163.
17. Friedrich, F.; Mendling, J.; Puhmann, F. Process model generation from natural language text. In Proceedings of the International Conference on Advanced Information Systems Engineering, London, UK, 20–24 June 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 482–496.
18. Leopold, H.; Mendling, J.; Polyvyanyy, A. Generating natural language texts from business process models. In Proceedings of the International Conference on Advanced Information Systems Engineering, Gdansk, Poland, 25–29 June 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 64–79.
19. Ivanov, D.; Sokolov, B.; Potryasaev, S. A dynamic model and an algorithm for supply Chain scheduling problem solving. In Proceedings of the 16th International Conference on Harbor, Maritime and Multimodal Logistics Modelling and Simulation, HMS 2014, Bordeaux, France, 10–12 September 2014; pp. 85–91.
20. Potryasaev, S.A.; Sokolov, B.V.; Ivanov, D.A. Control theory application to spacecraft scheduling problem. In Proceedings of the 2014 International Conference on Computer Technologies in Physical and Engineering Applications (ICCTPEA), St. Petersburg, Russia, 30 June–4 July 2014; pp. 145–146. doi:10.1109/ICCTPEA.2014.6893328.

21. Smirnov, A.; Teslya, N. Robot Interaction Through Smart Contract for Blockchain-Based Coalition Formation. In Proceedings of the IFIP International Conference on Product Lifecycle Management, Turin, Italy, 2–4 July 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 611–620.
22. Teslya, N. Industrial socio-cyberphysical system’s consumables tokenization for smart contracts in blockchain. In Proceedings of the International Conference on Business Information Systems, Berlin, Germany, 18–20 July 2018; Lecture Notes in Business Information Processing; Springer: Cham, Switzerland, 2019; Volume 339, pp. 344–355. doi:10.1007/978-3-030-04849-5_31.
23. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; p. 15.
24. Szabo, N. Formalizing and Securing Relationships on Public Networks. *First Monday* **1997**, *2*, 9.
25. Bano, S.; Sonnino, A.; Al-Bassam, M.; Azouvi, S.; McCorry, P.; Meiklejohn, S.; Danezis, G. SoK: Consensus in the age of blockchains. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies, Zurich, Switzerland, 21–23 October 2019; pp. 183–198.
26. Sokolov, B.; Ushakov, V. Model-Algorithmic Support for Abilities Calculating of Control System Based on Projection Operators. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 24–27 April 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 342–348.
27. Smirnov, A.; Sheremetov, L.; Teslya, N. Fuzzy cooperative games usage in smart contracts for dynamic robot coalition formation: Approach and use case description. In Proceedings of the 21st International Conference on Enterprise Information Systems, ICEIS 2019, Heraklion/Crete, Greece, 3–5 May 2019; Volume 1, pp. 361–370.
28. Teslya, N.; Potryasaev, S. Project Repository. <https://cais.ias.spb.su/gitlab/tnn/robot-coalition-blockchain> (accessed on 4 January 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).