Integrating Smart Contracts into Smart Factory Elements' Informational Interaction Model

Julia Lyakhovenko[a], Ilia Viksnin[a] and Sergey Chuprov[a]

[a]*ITMO University, Kronverksky Pr. 49, 197101, St. Petersburg, Russian Federation*

Abstract

The paper proposes a model for Smart Factory's elements informational interaction and organizing their functioning using smart contracts. An approach for modeling Smart Factory is based on the Enterprise Resource Planning organizational strategy, which assumes the division of the manufacturing process into modules, and is focused on continuous balancing and resource optimization. The proposed model includes a common Central Computer, modules for resource managing, agents responsible for operations, and task executors. The Central Computer receives a response from the External Environment, sends it to resource managing module, which determines the required amount of resources for the manufacturing operation and distribute tasks to the executors. To increase the production performance, two types of smart contracts are introduced: the former for defining a contract between the Smart Factory and the environment, and the latter for the production process organization. The integration of smart contracts into the production process allows to automate decision-making and control procedures, reduce the probability of manufacturing a low-quality product, and diminish the production costs.

Keywords

Multi-agent system, Smart contracts, Smart factory, Smart factory model

1. Introduction

Smart contract is a block of computer code in which the agreement on the transaction between the parties are formed. After the parties established the contract, it is maintained in the blockchain and comes into force. Such contracts are used in various fields such as finance and insurance, e-commerce, taxation and auditing, etc.

The integration of smart contracts and blockchain technologies into various economy sectors [1] is crucial for the Internet of Things (IoT) concept, which connects various physical objects with each other via Internet and provides their informational interaction.

Decision-making and execution control automation [2] in such an area as manufacturing accelerates the production process, reduces its costs, and diminishes the number of system vulnerabilities. The introduction of smart contract technology into the production system (hereinafter referred to Smart Factories) allows to decrease risks of manufacturing a low-quality product that does not meet the declared requirements, and ensures an increase in production processes performance and accuracy. The advantages of integrating smart contracts into the

Proceedings of the 12th Majorov International Conference on Software Engineering and Computer Systems, December 10–11, 2020, Online Saint Petersburg, Russia
lyakhovenko.kam@gmail.com (J. Lyakhovenko); wixnin@mail.ru (I. Viksnin); chuprov@itmo.ru (S. Chuprov)
0000-0001-7396-2831 (J. Lyakhovenko); 0000-0002-3071-6937 (I. Viksnin); 0000-0001-7081-8797 (S. Chuprov)

production process are their application transparency, immutability, self-executing property.

The contribution of this work is twofold. First, we propose a Smart Factory model, which is described in the Enterprise Resource Planning organizational strategy context. The model divides manufacturing process into modules, and is focused on continuous balancing and resource optimization. In addition, we integrate smart contracts into Smart Factory elements' informational interaction model, which allows to regulate the relations with the External Environment and between Smart Factory elements'.

2. Related Work

Internet of Things (IoT) access control is a critical issue. In [1], the authors propose an intelligent environment based on smart contracts, consisting of several contracts types: access control contracts (ACCs), one judge contract (JC) and one register contract (RC). In such a system, each ACC provides one access control method for the subject-object pair and implements it both statically checking access rights based on predefined policies, and dynamically, verifying access rights by checking the subject's behavior. JC implements an abnormal behavior scoring method to facilitate ACC dynamic check behavior by receiving abnormal behavior reports from the ACC, assessing that behavior, and returning an appropriate penalty. The RC records information on access control and abnormal behavior assessment methods, and provides functions (such as registration, update, and deletion) to manage these methods.

In [3] an intelligent logistics solution that includes smart contracts, logistics planning and asset health monitoring in supply chain management was proposed. A prototype solution was also implemented, demonstrating accountability, traceability, and responsibility for managing the supply of various parties involved in the logistics scenario. The proposed solution uses a smart contract system (SCS) that provides a recommended list of suppliers for a specific item to be purchased. Once a list of suppliers is selected, the SCS sends notifications to the appropriate suppliers and begins the negotiation process between buyer and supplier regarding conditions. After the negotiation process is complete, a smart contract is established and a purchase order is created. SCS receives periodic updates from the planner and condition monitoring modules that provide information on the current purchase order status. These updates are used by SCS to verify that the contract terms are met in accordance with the agreement. In the event of any irregularities, SCS initiates the appropriate action specified in the contract, such as a certain penalty or even cancellation of the purchase order.

3. Smart Factory Model

In this paper we consider centralized factory elements informational interaction model. Under a centralized factory we mean a production model that has one common center responsible for communication with the External Environment and production processes control. The Smart Factory model description is formulated in the context of an Enterprise Resource Planning (ERP) organizational strategy [4, 5], which assumes the division of the manufacturing into modules, and is focused on continuous balancing and resources' optimization. There are three main groups of modules: finance, personnel, and operations. Our proposed model has one common center and several control modules: a resource management module, an executor management module, and an operations management module. To characterize a model, a multi-agent approach was applied, which considers a system as a set of interacting intelligent agents.

The following types of agents were identified:

- Central Agent - Central Computer;
- Modules Agents: Resource Management Module Agent, Executor Management Module Agent, and Operations Management Module Agent (Comp1, Comp2, Comp3 respectively);
- executors.

There is also some abstract wireless communication channel in the system. Figure 1 demonstrates our proposed Smart Factory informational interaction model overview.

Central Agent is the Central Computer, which communicates with the External Environment (for example, with a customer), verifies the information received from the control modules, and sends this information to the DataBase. The DataBase stores information about orders received by the Smart Factory, the work of system agents and manufactured products. Management Modules agents determine the available system resources, resources required for the production, and perform task-distribution procedures. Executing agents have m levels, for each level q access parameter is defined, i.e. set of executing agents: $Executors = \{(a_{1_1}|q_1), (a_{1_2}|q1), \ldots, (a_{m_1}|q_m), (a_{m_n}|q_m)\}$, where q is the agent's access parameter: $0 \leq q \leq 1$, that distributes agents by access levels and determines the available functionality. The following parameters are defined for each agent-executor:

- a set of functions: $F_j = \{f_1, f_2, \ldots, f_l\}$, which depends on the agent's access level and parameter;
- agent's resource set: $R_{ij} = \{r_1, r_2, \ldots, r_k\}$, where i relates to the agent $a_i \in Executors$, and j represents his level;
- the remaining time of executor a_{ij} is $t_{w_{ij}}$, if $t_{w_{ij}} \approx 0$, then the agent a_{i_j} is redirected from the set of $Executors$ to the $Empty = \{a_{ij} \in Executors : t_{w_{i_j}} \approx 0\}$, and $Executors = Executors \setminus Empty$

Each $a_{i_j} = \{status_{i_j}, location_{i_j}, R_{i_j}\}$, where $status_i = 0$ if the agent is busy, and 1 if the agent is free, and $location_{i_j}$ is the agent's coordinates. The information space can be described as $I = I_{fun} \cup I_{rules} \cup \ldots \cup I_{client}$, where I_{fun} is the information on Smart Factory's functionality, I_{rules} is the Smart Factory's rules, and I_{client} is the information on Smart Factory's consumers.

Central agent has global knowledge on the information space. The resource management module possesses information on the resources. The agent-executors management module has information on the agent's location and their status. The operations control module has information on the different levels of executing agents' functions, ways of transmitting information over the communication channel such as encryption, transmission protocols, etc. The combination of these modules constitutes a certain computer Shop.

The manufacturing production can be represented as $PR = \{pr_1, pr_2, \ldots, pr_s\}$, after converting each product into some function by the Central agent: $pr_i = fun(A_i, F_i, R_i, I, t_{en})$, this information is sent to the Shop, where, according to the provided function, resources, and

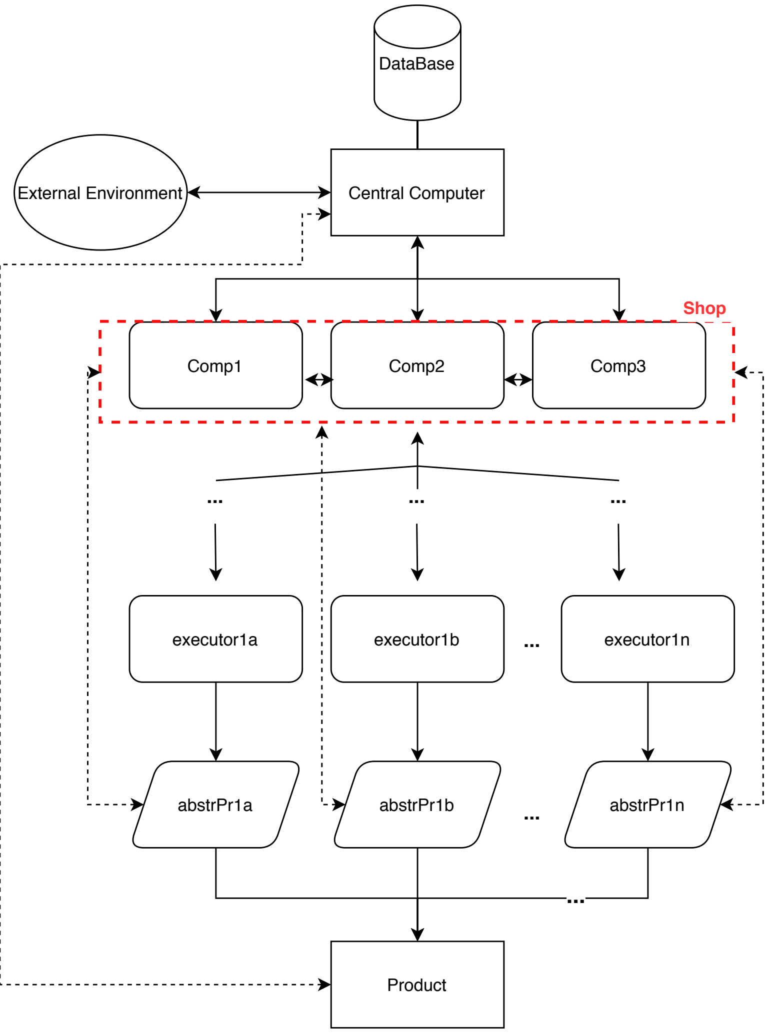

Figure 1: The schematic representation of the proposed Smart Factory model.

the required time, tasks are distributed among the executors agents. The production process is divided into several tasks: $pr_i = \{ts_1, ts_2, \ldots, ts_d\}$, where $ts_j = fun(pr_i)$ and $\sum ts_j = pr_i$, which means that all tasks are required to be performed to manufacture the product.

Further, at each executing agents level, a task is performed, for which the access level and functionality are defined. After that, the executing agents proceed to perform the assigned tasks, the results of which are: $abstrProduct_{1a}, abstrProduct_{1b}, \ldots, abstrtProduct_{1n}$. Further information on abstract products (represented by (1)) is sent to the Shop, where it is verified for the correctness of their assigning to the executing agents. All verified abstract products are assembled at pr_i. After that, information on the product is sent to the Central Computer, where it is verified for compliance with the product requirements. If the verification procedure passed successfully, the information on the product is sent to the DataBase, and the product is sent to the customer.

$$I = I(abstrProduct_{1a}) \cup I(abstrProduct_{1b}) \cup \ldots \cup I(abstrProduct_{mn}) \quad (1)$$

4. Our Approach to Smart Contracts in Smart Factory

For the centralized factory model, we introduce two smart contracts' types. The former assure the relationship between the External Environment and the Smart Factory, the latter is used to organize the work inside the factory.

The response for product pr_i manufacturing comes from the External Environment. Then, the Central agent determines the required resources and functionality: $pr_i = fun(A_i, F_i, R_i, I, t_{en})$. If the pr_i manufacturing is possible, a smart contract is composed between the Factory and the External Environment: $Contract_i = fun(Central, pr_i, t_{en})$, where t_{en} is a producing time limit, which is determined by the External Environment, $Central$ is a function that defines the resources and functionality, required to be performed by the factory: $Central = fun(A_j, F_j, R_j, I, t_{wj})$. However, there are several conditions for the contract generation process:

- if $Central \to 0 : Contract_i = False$;
- if $t_{wj} \in Central < t_{en} : Contract_i = False$;
- $Central \geq pr_i$.

Information on the established contract is sent to the DataBase: $DB = \{Isc_1, Isc_2, Isc_3, ...\}$, where $Isc_i = I(Contract_i)$. After the pr_i manufacturing response is transferred from the Central agent to the Shop, the Shop split the production procedure into certain tasks: $pr_i = \{ts_1, ts_2, \ldots, ts_d\}$, $ts_j = fun(pr_i)$. For each task, the access level q_{en}, resources amount R_{en} and functionality are defined F_{en}: $I(ts_j) = \{F_{en}, R_{en}, q_{en}\}$.

In the next step, the Shop forms internal smart contracts: $contract_i = fun(Executors_a, F_{ex}, R_{ex}, ts_j, t_{en})$, where $Executors_a \subset Executors$ is set of task executors agents, and F_{ex}, R_{ex} are functionality and resources amount of these agents. In addition, the following contracts' conditions are introduced:

- if $ts_i \geq fun(Executors_a, F_{ex}, R_{ex}) : contract_i = False$;
- $R_{ex} \to 0 : contract_i = False$;
- $I(ts_i) \leq F_{ex}, R_{ex}, q_{ex}$;
- $tw_{\geq} t_{en}$.

If one of the contact's conditions does not met, the contract is canceled.

When the contract is established, information on it is sent to the factory's DataBase: $DB = \{Icon_1, Icon_2, Icon_3, \ldots\}$, where $Icon_i$ is represented by (2). That is, knowledge on the contract i is linked to the knowledge on the previous contract $i - 1$. This blocks' sequence forms a chain of related information. Information on the first contract $contract_0$ is defined as: $Icon_0 = I(contract_0)$.

$$Icon_i = I(contract_i) \times I(contract_{i-1}) \quad (2)$$

5. Conclusion

In this paper, we proposed a novel approach for Smart Factory elements informational interaction using smart contracts. The model is based on a multi-agent approach and ERP organizational strategy, which divides the production process into several modules and is focused on continuous balancing and resources optimization. Two smart contract types are introduced: the former for generating a contract between the Smart Factory and the External Environment, and the latter for manufacturing organizing inside the Smart Factory. The smart contracts integration into the production process allows to automate decision-making and execution control, reduce the probability of manufacturing a low-quality product, and decrease production costs.As future plans, we intend to enhance the proposed Smart Factory elements' informational interaction model and integrate other manufacturing organization means into it, as well as conduct an empirical study to assess the performance gain with smart contracts.

References

[1] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, J. Wan, Smart contract-based access control for the internet of things, IEEE Internet of Things Journal 6 (2018) 1594–1605.

[2] M. Wohrer, U. Zdun, Smart contracts: security patterns in the ethereum ecosystem and solidity, in: 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), IEEE, 2018, pp. 2–8.

[3] E. J. Umble, R. R. Haft, M. M. Umble, Enterprise resource planning: Implementation procedures and critical success factors, European journal of operational research 146 (2003) 241–257.

[4] M. Al-Mashari, A. Al-Mudimigh, M. Zairi, Enterprise resource planning: A taxonomy of critical factors, European journal of operational research 146 (2003) 352–364.

[5] S. Rouhani, R. Deters, Security, performance, and applications of smart contracts: A systematic survey, IEEE Access 7 (2019) 50759–50779.