Design of batch manufacturing enterprises in the context of Industry 4.0

Valery Taberko
and Dzmitry Ivaniuk
and Valery Kasyanik
JSC «Savushkin Product»
Brest, Republic of Belarus
id@pda.savushkin.by,
val@pda.savushkin.by

Vladimir Golovko
Brest State Technical University
Brest, Republic of Belarus
gva@bstu.by

Natalia Guliakina
and Kirill Rusetski
and Daniil Shunkevich
and Alexander Boriskin
and Natalia Grakova
Belarusian State University
of Informatics and Radioelectronics
Minsk, Republic of Belarus
{guliakina,rusetski,grakova}@bsuir.by,
shunkevichdv@gmail.com,
coloss_000@mail.ru

***Abstract*—This paper presents an evolution of an ontology-based approach to designing batch manufacturing enterprises. According to Industry 4.0 approach, instead of isolated view of a manufacturing process inside a single enterprise this new approach encompasses related business entities as well - raw material suppliers (e.g. dairy farms) and large-scale consumers (e.g. stores or retail chains). Special attention is paid to logistics processes: a short description of fundamental logistics processes of cottage cheese production is provided, as well as subject domain structure of logistics and an example of formal specification of emergency logistics situation. It is shown that multiagent industrial control system with agents interacting via shared memory is compliant with design principles of Industry 4.0 approach. Standards formalization topic is touched upon as well. PFC, a graphical procedural model specification language, formalization is discussed. PFC is specified in ISA-88.02 standard. Graphical language formalization allows industrial control system users to communicate to it in a unified manner using diagrams that are widely understood by engineering specialists. This paper also outlines an agent-oriented approach to robot interaction within industrial robotic complexes based on shared semantic memory interaction mechanism.**

***Keywords*—integrated industrial control automation, logistics process, ontology-based enterprise model, Industry 4.0, cyber-physical system, ontology, knowledge base, OSTIS technology.**

I. Introduction

This article considers the further automation development of the batch manufacturing enterprises by the example of JSC "Savushkin Product". It consists in the transition from consideration of manufacturing processes, occurring within a specific enterprise, to the consideration of the full product cycle – from a store application to its execution (satisfaction of the consumer's request). This process, apart from the enterprise, as subjects include dairies, logistic services, shops and others. Also, even within the framework of one enterprise, disparate systems are used today (several SCADA systems, a transport management system, a warehouse management system, etc.). Coupling of such systems and maintenance of information consistency in them is carried out manually (or omitted). Now there is a need to automate information coordination and to provide, where appropriate, the interaction of devices at different stages of the product cycle. The existence of such requirement has led to the forthcoming Industry 4.0 in Germany and its analogs in other countries.

The objective of this paper is therefore an evolution of ontology-based model of batch manufacturing enterprise which widens the scope of production stages to include production processes that are happening outside of an enterprise, in accordance with the Industry 4.0 initiative. External logistics processes, such as milk delivery from dairy farm to dairy factory, or final product delivery to customers, can be used as examples.

A. *Industry 4.0 and cyberphysical systems*

The concept Industry 4.0 was formulated in Germany in 2011. It means creating and implementing production-ready cyberphysical systems, as well as using Internet of Things (IoT) and Internet of Services (IoS) in manufacturing processes[1]. Note that this term is specific to Germany, and is rarely used outside. There are concepts similar to Industry 4.0 in other countries: Smart factory in Netherlands, Future Factory in Spain, Industrial Internet (of Things) in the USA.

Industry 4.0 design principles are outlined below [2]:

- **Interoperability.** CPS and humans are connected over the Internet of Things (IoT), Internet of People (IoP) and the Internet of Services (IoS).
- **Virtualization.** CPS can monitor physical processes. Sensor data are linked to virtual plant models, which include the current state of all CPS. In case of failure a human can be notified and provided with all necessary information and, hereby, supported in handling the decision-making.
- **Decentralization.** The rising demand for small batches of custom-ordered products makes it increasingly difficult to control manufacturing systems centrally. CPS can have

computers embedded in them to enable them to make decisions on their own, and only delegate tasks to a higher-level equipment in case of failure. Nevertheless, for quality assurance and traceability it is necessary to have centralized control of the system. For example, RFID tags can "tell" machines which working steps are necessary, thus eliminating the need for the centralized control of this part of the small or individual batch production process.
- **Real-Time Capability.** For organizational tasks it is necessary that sensor data is collected and analyzed in realtime. In case of an equipment failure its task can be rerouted to another piece of equipment.
- **Service orientation.** The services of companies, CPS, and humans are available over the IoS and can be utilized by other participants. Services can be offered both inside and outside the company. CPS can offer their functionality, for example, as a set of web services. It allows for composition of production process from smaller operations according to a customer specification encoded on an RFID tags, for example.
- **Modularity.** CPS have to be flexible to easily adapt to changing requirements (e.g., seasonal fluctuations or changed product or production environment characteristics) Adaptation can be done by replacing or expanding individual modules of the system. Module compatibility requires standardized software and hardware interfaces, so that new modules are identified automatically and can be utilized immediately via the IoS.

CPS is a collection of intelligent, easily integrated physical components with built-in computational resources that closely cooperate and monitor changes in their environment [3].

To build a CPS it is necessary to integrate computational resources and technical processes. Sensors, manipulators, information and control systems should interoperate at all stages of the production including those outside of a particular factory [4], [5], [6], [7], [8]. It needs to be said, though, that implementation of a new industrial control system, including intelligent ones, should be based on resources that the enterprise already possesses [9].

Need for integrated automation of complex processes that require coordinated interaction of multiple services and technical equipment drives creation of such systems. From now on we will discuss enhancement of the industrial control level of the batch manufacturing enterprise in terms of designing a CPS responsible for producing cottage cheese "Khutorok" at JSC "Savushkin product".

Such CPS should offer informational support and industrial control automation throughout the entire cottage cheese "Khutorok" production process – from milking a cow to delivering final product to the store. This process can be divided into the following main stages:

1) gathering milk at the farm;
2) milk delivery from the dairy farm to the dairy factory;
3) milk processing at the factory, cottage cheese production;
4) forming and packing;
5) final product delivery to the customer (shops, retail chains, etc.).

B. Problems in integrated enterprise control systems development and the proposed approach to solving them

Main problem with integrated enterprise control systems development lies in integration of its various components and facilitation of their interoperability. It can be solved the traditional way, by developing communication layers between heterogenous components of the system (interfaces, protocols, etc.). On the one hand it leads to considerable overhead required to develop them, on the other hand it complicates system architecture, which leads to increased costs of its maintenance and further development. Continuous evolution of production technologies at various stages and expansion of production itself requires industrial control system to be flexible, i.e. able to be easily extended by various components. Existing components should be modifiable when possible or required. To solve this problem we propose to extend the original ontology-based approach to design of batch manufacturing enterprises [10]. Enterprise is viewed as an integrated multiagent system, within which:

- all information is integrated within unified informational space (enterprise knowledge base stored in the semantic memory);
- all participants (people, robots, various integrated production systems etc.) are interpreted as agents that are working with this shared knowledge base; It means that (a) they are monitoring the knowledge base for the situations they can handle and (b) they specify results of their work in the knowledge base, so that this information is available for other agents to analyze. This approach reduces production process management to proper specifications of tasks in the shared knowledge base – with time-frames, priorities, assignees, etc.
- knowledge base has hierarchical structure, i.e. is a hierarchy of subject domains and corresponding ontologies.
- multiagent system is a hierarchical system in itself - agents can form infinitely nested collectives, since particular collective as a whole can be a member of another collective. For example, a group of robotic systems can be logically (or even physically) joined to form an integrated robotics system which can solve certain class of problems.

Multiagent system over a shared knowledge base implemented using ***OSTIS technology*** is therefore Industry 4.0-compliant and can be interpreted as a CPS:

- **shared knowledge base** implements **interoperability** of people, sensors, and equipment, serving as an intermediary of an interoperation, **virtualization** – knowledge base contains a representation of an enterprise model and environment with necessary level of details, **service orientation** – every participant of a production process (agent) is specified within the shared knowledge base

including its functionality (services provided), **modularity** – it hosts a library of reusable and interoperable components.
- **multiagent approach** implements principles of **decentralization** by the definition of a multiagent system [11], **realtime analysis and reaction** – agents monitor enterprise knowledge base state and activate in response to certain situations (including emergencies).

This approach offers several advantages, such as:

- there is no need to develop interoperability tools for system components (human-robotic interaction, human-human interaction tools, etc.) due to their interaction via shared memory
- since all agents interact via shared memory, in general, physical implementation of the agent does not matter to the system. Therefore, gradual replacement of manual labor by automated systems or the improvement of such systems does not require changes to the industrial control system in general;
- due to the use of a shared knowledge base and associative search in such knowledge base, any production process participant has access to all the information at any time, as needed, not only to a limited number of predetermined fragments, increasing the number of which may incur additional overhead costs. Thus, monitoring various processes becomes easier, and the answers to user questions can be found faster. User requests can be elaborated in numerous ways;
- information stored in the knowledge base can be rendered differently for various categories of users, and while the information itself remains unchanged, only the mechanism of displaying it will change. Therefore, there is no need to duplicate information;
- since all production processes are specified and managed via the knowledge base, making changes to such processes generally boils down to making changes in the knowledge base and replacing the corresponding equipment, if necessary. At the same time, the overhead costs for reprogramming the components of the system, and for facilitating the interaction between them, and for facilitating the interaction between them are substantially reduced;
- specification of all production processes in a shared knowledge base provides diverse options for their automatic analysis, including continuous monitoring of current processes, automatic detection and elimination of emergency situations, optimization of current processes, automatic planning of future processes, etc.

C. Architecture of the proposed system

Proposed system is based on the ***OSTIS Technology*** and according to it consists of a knowledge base, knowledge processing machine, and user interface In general, industrial control system knowledge base consists of [10]:

- ontologies of the industry-specific standards, such as ISA-88 [12]
- enterprise models based on these ontologies (e.g., physical, procedural and process models for ISA88)
- enterprise improvement ontologies, that formalize principles of improving and adapting an enterprise to changing conditions
- tools of collaborative development of enterprise knowledge bases and knowledge processing machines
- industrial control systems user interface models
- information service model for various user classes
- enterprise knowledge representation models that allow to specify it in all of the necessary aspects:
  - enterprise knowledge management model [13];
  - ontology-based enterprise model [14];
  - multiagent enterprise model [11];
  - enterprise situational control model [15];
  - business process re-engineering model [16].

Previous paper [10] discussed the formalization of standards, in particular ISA-88, in the form of a family of ontologies. Several fragments of enterprise models that were formalized using these ontologies, were shown. This paper also touches upon the ISA-88 standard, but focuses on the enterprise procedural model specification language – PFC, which is described in Part 2, Chapter 6 of this document [17]. In addition, the article shows evolution of an ontology-based enterprise model for a formal description of processes occurring outside the enterprise, in particular, logistical ones. Enterprise CPS knowledge base should contain following models, among others, to adequately describe manufacturing process:

- models of process cell description languages
- process cell models described using these languages
- logistics process models
- enterprise robot interaction model

The model of logistics processes is necessary at all stages of production to describe the internal (inter-shop logistics, warehouse logistics) and external logistics processes (cooperation with milk supply farms and retail partners). Models of process cell description languages and models of cells described with them are used at the third and fourth stages mentioned in the introduction, to formalize the production process and the structure of the equipment used for this. The interaction model of industrial robots specifies the physical model of the enterprise. The concept of the robot has wider interpretation in this context, which includes equipment modules and their complexes that perform their tasks with minimal human intervention, if any.

II. Implementing models, that are used to describe the enterprise and its processes

A. Language model for process cell description languages

For the convenience of the enterprise personnel operation with the system of complex automation, it is necessary to ensure, on the one hand, the ability of the system to interact with users in convenient ways (including using various graphic languages, limited natural language and voice messages), on the other hand – to provide the possibility to add new language means to the system, for example, new graphic languages.

Each intelligent system operates with a knowledge base in the internal language, and the dialog is implemented as a message exchange between the user and the system. To facilitate such dialog, it is necessary to convert certain knowledge base fragments into their external representation. Such representation can either be universal or specialized.

The universal external language for message exchange we will call the external language for message exchange, which allows to describe knowledge of any kind. Such is Semantic Computer Code (SC-code) and all its representations.

Translation from the internal language to external and back is organized in such a way that the translation mechanisms do not depend on the external language. In order to implement a new specialized language, in this case it will be necessary only to describe its syntax and semantics, while the universal translation model will not depend on this description.

Every language is characterized using three primary aspects. Each one of these aspects is described in the corresponding ontology.

- *Ontology of language semantics*
- *Ontology of language texts*
- *Ontology of language rules*

Language semantics ontology implies the choice of a set of uniquely defined entities that are understandable at the associative level and which carry a certain meaning.

Language texts ontology researches the syntactic structures that are images, symbolic representations of the language entities. Number of these kinds of images (symbols) is not limited and depends on the context being used.

The ontology of language rules is directed to the consideration of the rules of the language specifies unambiguous correspondences between the set of entities (the alphabet of the language) and the set of images (file signs) used to translate texts into and out of the intelligent system memory and also to visualize these texts.

The mechanism of translation is provided due to the presence in the system of a set of receptor and effector agents [18] in the mode of permanent exchange of messages between the user and the system. This message exchange mechanism is as follows:

1) The user writes some information with the editor of one of the specialized external languages.
2) Receptor agents fix the fact of the translation start of the written syntactic structure.
3) Internal agents use a set of rules to transform the syntactic structure into a sequence of sc-elements that constitute a fragment of a semantically connected sc-text that is unambiguously interpreted in the system's memory.
4) If the user makes changes in the resulting sc-text, then the reverse process occurs: the correspondence between the entity signs and their images is established, as a result of which the syntactic structure in the selected language is displayed.

The semantics of any language implies the introduction of a set of strictly defined entities sufficient for writing texts that represent a sense for the user or machine (system). Texts of a language are understood as syntactic structures that are images of the language entities. Finally, the rules of the language specify unambiguous correspondences between the set of entities (the alphabet of the language) and the set of images (file signs) used to translate texts into and out the intelligent system and to visualize these texts.

In the process of each language description, it is possible to identify certain aspects that are common for all languages or a particular family of languages. Research in this area are aimed at justifying a certain metalanguage, which defines the structure for describing the majority of existing languages. This meta-language will give impetus to the development of natural-language interfaces and will allow to introduce algorithmic precision into the linguistic aspects of any language.

The technology of cottage cheese production can be described in accordance with the ISA-88 standard. In the context of automated production and the ISA-88 standard, the following specialized external languages are distinguished: the procedural model description language (PFC) and the physical model description language (P&ID). With the use of the PFC language, a fragment of the production cell for the production of "Khutorok" cottage cheese will be described, which will be considered below.

The PFC language is defined in Chapter 6 of ISA-88.00.02 and is intended to describe recipes with complex procedures, involving parallel steps and conditional branching. PFC diagrams represent procedural logic using a set of icons connected by directional connections indicating the order in which procedural elements are executed.

The alphabet of the PFC language includes the following elements:

- procedural elements – elements of a procedural hierarchy (phases, operations, etc.);
- additional elements – elements responsible for allocation, synchronization and transfer of the resources within procedure (allocation element, synchronization element, etc.);
- structures represented in the form of classes of temporal entities that specify order.

B. Model of production cell of cottage cheese "Khutorok"

As an example of the procedural model use, a production cell of the "Khutorok" cottage cheese produced by "Savushkin Product" enterprise will be used. This cell reflects the stage of the milk processing in the plant using the example of a specific product manufacturing. The structure of the "Khutorki" project is presented on Listing 1:

Project "Khutorki"
=> *inclusion**:
    master recipe
    => *inclusion**:
        • *recipe procedure*

- *equipment procedure*
  => *inclusion**:
    - *"Milk whey separation" operation*
    - *"Milk whey pumpdown" operation*

Listing 1. Procedural hierarchy specification for "Khutorki" project

Master recipe describes the process from processing milk mixture to curd mass packing. The recipe and equipment procedures focus on the production of curd mass as is the master recipe. Finally, hardware procedure focuses on the operation of separating and pumping out the milk whey.

Fig. 1 shows the PFC representation of the procedure (PFC alphabet was discussed earlier).

Procedural model fragment that corresponds to the PFC chart in Fig. 1 is shown in Fig. 2. Entities and relations used in this structure will be explained in listings 2-3 and accompanying text.

PFC element
=> *inclusion**:
- *structural element*
- *procedural element*

Listing 2. PFC element classification

Structural element is a PFC element which in conjunction with several procedural

Procedural element is an element of procedural hierarchy which includes phases and operations.

execution order*
<= *subdividing**:
{
- *implicit transition**
- *explicit transition**
}

Listing 3. PFC transition element classification

Implicit transition* is a binary relation, the first component of which is a procedural element, after which execution of the procedural element, which is the second component, will begin.

Explicit transition* is a binary relation, the first component of which is a procedural element, after which execution of the procedural element, which is the second component, will begin, after certain condition evaluates to True.

Transition condition* is a binary relation, the first component of which is the instance of an explicit transition* relation, the second is the structure containing the expected result of the procedure.

Objects that are studied in the Ontology of PFC texts and an example of a corresponding syntactic structure will be shown in listing 4 and Fig. 3, accordingly.

PFC element image
=> *inclusion**:
- *resource allocation element image*
- *synchronization element image*
- *procedural element image*
- *procedure nesting indicator image*
- *procedure execution element image*
- *directed link image*

Listing 4. Specification of PFC element images

Resource allocation element image – the image of an oval whose ***identifier**** is the resource specification.

Synchronization element image – image of a rectangle adjacent to the image of a linear primitive, which is the height of the rectangular primitive of the procedural element.

Procedural element image – image of a rectangular primitive, the number of selected right angles indicates the position of the procedural element in the procedural hierarchy.

Procedure nesting indicator image – a plus sign inside the right-hand, separated from the observer, the right angle of the image of the procedural element and touching the boundary of this selected corner.

Procedure execution element image – image of a graphic primitive associated with the phase of executing a procedural element.

Directed link image – image of a linear primitive incident to images of PFC language elements.

Image caption* – a binary relation the first component of which is the sign of the image of the procedural element, and the second is the sign of the file containing some textual explanation to the image of the procedural element.

Any rule in the **Ontology of PFC language rules** is a *correspondence* defined on *atomic formulas*. Semantic and syntactic aspects of entity identification rule are shown in Fig. 4.

C. Ontological model of the cottage cheese production logistic chain

Any production task can be considered as a complex logistical task. However, logistical processes are not limited to the scope of production shops and even the enterprise – they also cover delivery services and interaction with suppliers of raw materials and stores.

Logistical chain of cottage cheese "Khutorok" production includes the following stages:

- Dairy farm
  - Cow
  - Tank
- Dairy plant [production site]
  - Milk truck
  - Finished product shop [production cell]
    - Acceptance post
    - Acceptance tank
  - Soft cheese and cottage cheese production shop [production cell]
    - Milk storage tank
    - Coagulator
    - PFU
    - Cooler
  - Finished product shop [production cell]

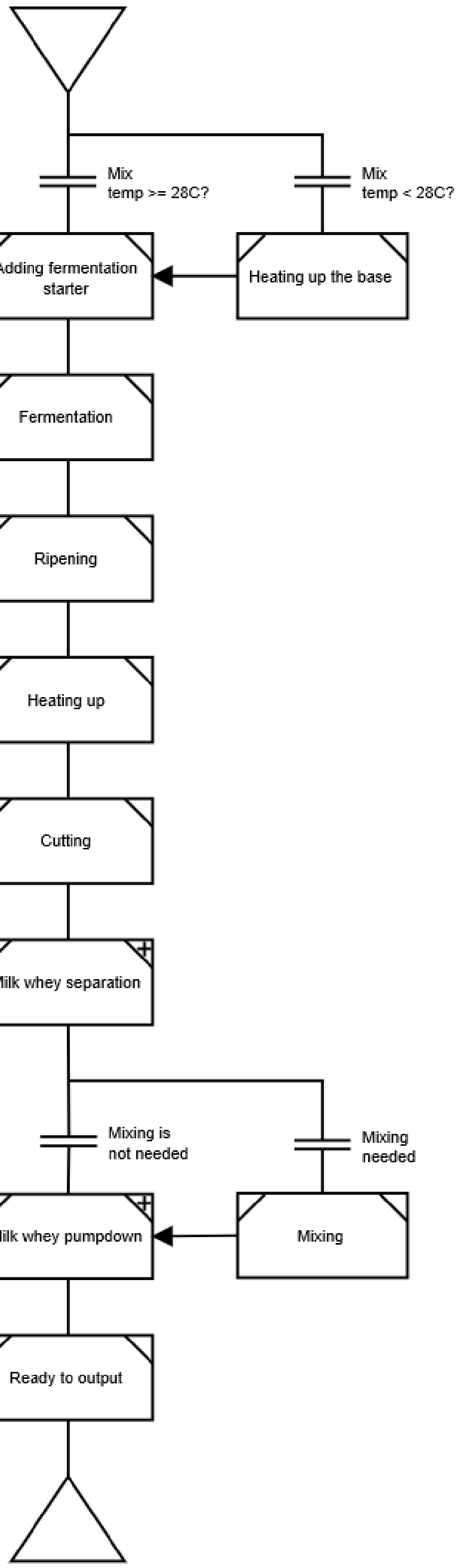

Figure 1. Equipment procedure for producing cottage cheese

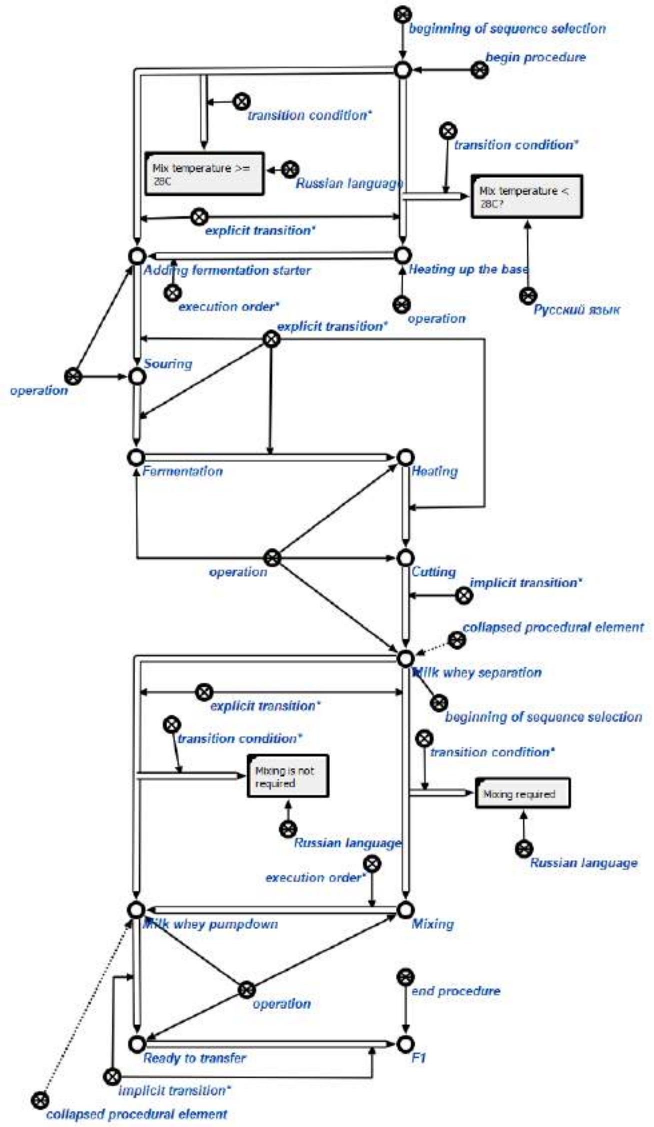

Figure 2. SCg representation of cottage cheese production equipment procedure

- ∗ Box
- ∗ Pallet
- ∗ Storehouse
- ∗ Set-up area Truck

- Store
  - Store warehouse
  - Store shelf
  - Customer

Fig. 5 schematically represents the logistics chain.

Briefly consider how today the main tasks within this chain are being solved.

Milk after milking on a dairy farm is collected in a special container (tank) and cooled before pumping into milk truck. The laboratory on the farm conducts organoleptic, chemical and other tests to determine the milk quality. Information about the samples is recorded in the company's accounting system, for example, 1C Enterprise.

Further, milk is pumped into milk truck, which carries it directly to the dairy plant. In order to minimize delays in the way and to avoid damage to milk, the movement of the milk truck is monitored at the plant. To solve this problem, vehicle position monitoring tools are used – TMS-systems, OpenStreetMap maps, information from the GPS-navigator of the car and the drivers mobile phone. In case of inaccessibility of GPS datalogist can directly contact the driver by phone to clarify it position.

After arrival at the milk plant, the milk again subjected to laboratory tests, the results of which are entered into the enterprise accounting system. Then it is determined in which milk storage tank and from which post milk to be pumped. Milk truck is sent to the appropriate post, where the acceptance operator initiates the milk transfer using the SCADA-system for process control and enters into the enterprise accounting system the necessary data for input raw material accounting.

After pumping milk in a tank, the milk is cooled. The acceptance operator, in coordination with the opera-tor of the hardware shop and the operator of the curd shop, prepares the mixture (using the pasteurizer) and feeds it to the desired curd shop coagulator (several SCADA systems, each operator uses its own project). The masters of the corresponding shops also keep records about intershop movement of mate-rial values using enterprise accounting system.

From the mixture in the curd shop coagulator, the operator prepares the curd mass, controlling the process by means of the SCADA system, and then supplies it for forming to the PFU. Operators of the filling line or robots shift the formed product into a consumer packaging – polyethylene packaging. Packed cottage cheese is labeled, cooled and the operators (or robots) fit in boxes, the boxes are stacked on pallets and through the conveyor get to the automatic warehouse, that is managed by a WMS (warehouse management system).

The pallet from the automatic warehouse along the conveyor is delivered to the set-up zone that is also managed by the WMS. There storekeepers or robots carry out loading of machines that deliver products to customers. At this time, the masters record the shipment of finished products in the enterprise accounting system.

Machines deliver products to specific customers, such as shops, retail chains, etc. Logistics monitor for delivery to the buyer product, using the same monitoring tools as in the milk delivery from farm to factory – TMS-system, OpenStreetMap maps, information from the vehicle GPS-navigator, GPS-navigator from driver's mobile phone, checking calls to the driver, and so on.

Thus, the logistics process for the "Khutorok" cottage cheese is rather complicated – many services and specific people are involved, about a dozen different software tools are used, the consistency of information in which is often supported manually by operators calling, manually entering information into accounting systems, etc.

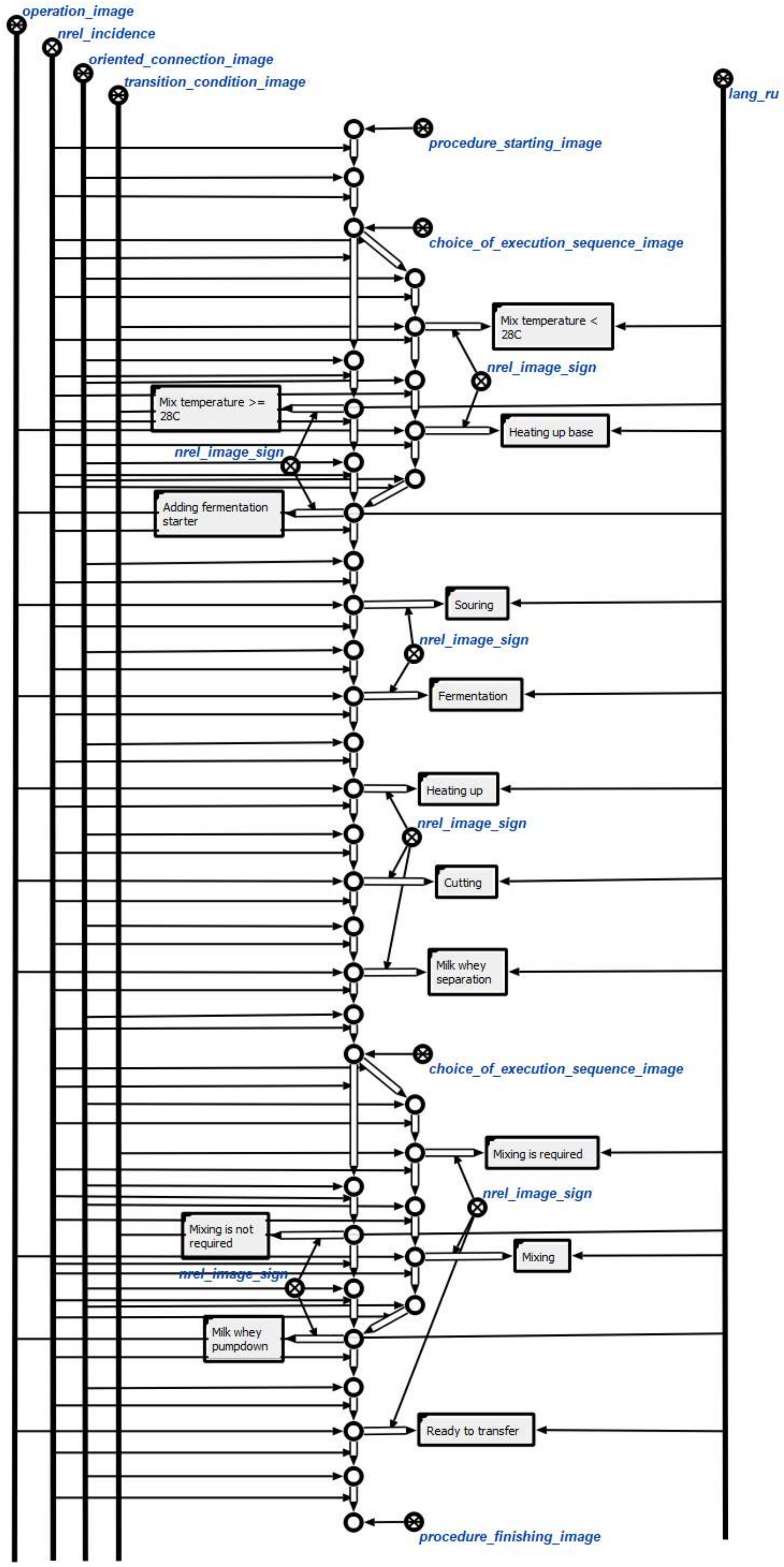

Figure 3. Syntax structure of the PFC diagram of the equipment procedure of the cottage cheese

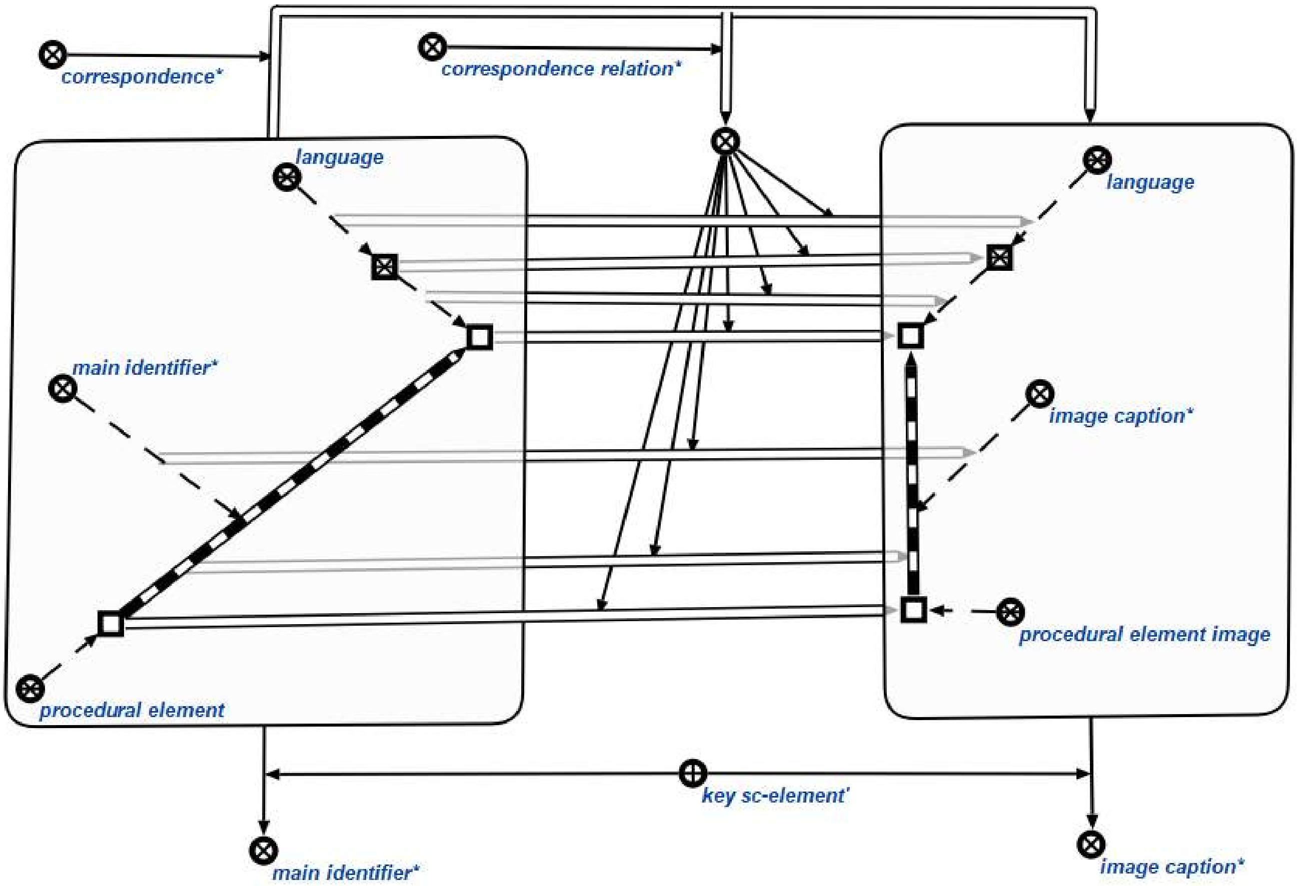

Figure 4. Translation rule example

So, even some parts of the logistics process, for example, warehouse management, include several subprocesses (business processes of the second level) [19]. You also need to take into account the restrictions on the production process duration (a few hours) and selling (a few days) of the manufactured products. The above description omits planning processes for short, which involve additional personnel and software.

In accordance with the ontological approach to the design of the enterprise, it is necessary to form a hierarchical system of subject domains and their ontologies to describe the logistic aspect of the enterprise's activity. The structure of the relevant sections of the knowledge base is shown in Listing 5.

Section. Subject domain of logistics
*<= section decomposition**:
{
- *Section. Subject domain of logistics processes*
- *Section. Subject domain of routes*
- *Section. Subject domain of logistics process participants*
  *<= section decomposition**:
  {
  - *Section. Subject domain of customers*
  - *Section. Subject domain of suppliers*
  - *Section. Subject domain of personnel*
  - *Section. Subject domain of transport*
  - *Section. Subject domain of orders*
  - *Section. Subject domain of logistics documents*

  }

}

Listing 5. Section structure of the logistics knowledge base

Listing 6 shows the examples of structural specifications of some subject domains.

Subject domain of logistics processes
∈ *key sc-element'*:
Section. Subject domain of logistics processes
∋ *maximum class of research objects'*:
logistics process
∋ *researched relation'*:
- *working hours**
- *start time'*
- *end time'*
- *execution time**
- *working days**

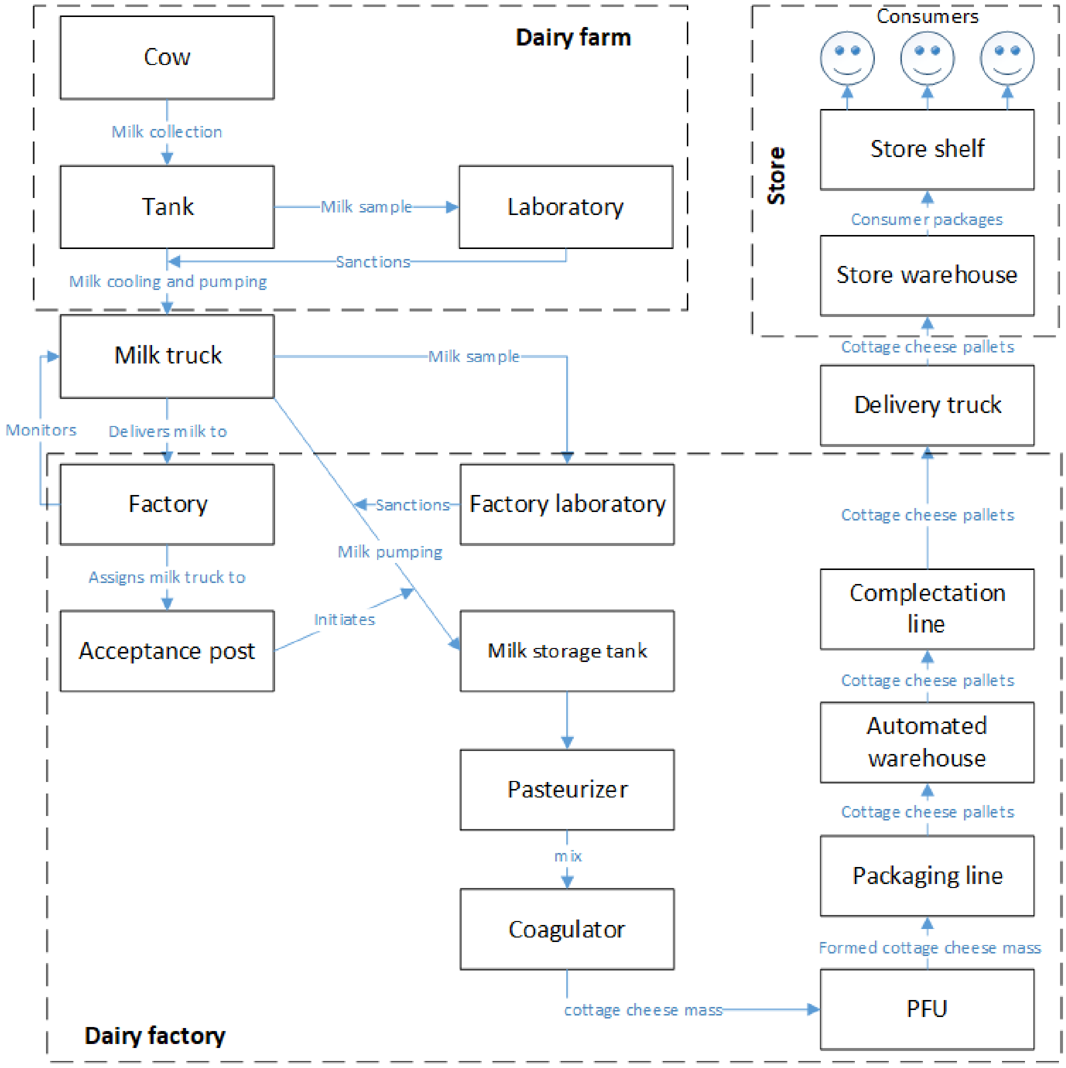

Figure 5. Logistics chain of cottage cheese production

∋ *non-maximum class of research objects'*:
- *truck arrives at a warehouse*
- *processing of the customer's request*
- *planning of shipments*
- *receiving customer orders*
- *route calculation*

Subject domain of routes
∈ *key sc-element'*:
Section. Subject domain of routes
∋ *maximum class of research objects'*:
route
∋ *researched relation'*:
- *destination point'*
- *departure point'*

Listing 6. Subject domain specification examples

Based on the developed system of subject domains, an example of a logistical situation involving two vehicles was formalized, one of which was involved in an accident.

Listings 7 and 8 depict formal description of the situation.

Truck 1
=> *coordinates**:
[52.19206, 25.266405]
∈ *participant'*:
- *Car accident*
- *Glaze*

∋ *departure point'*:
"Ruzhany-agro" farm
∋ *destination point'*:
JSC Savushkin Product
∈ *truck*

Truck 2
=> *coordinates**:

[52.265865, 23.967364]
∋ *departure point'*:
JSC Savushkin Product
∋ *destination point'*:
"Vasilishki" farm
∈ *truck*

Listing 7. Example of logistics situation participants specification

Listing 7 shows specification of the participants of the logistics situation. Note that situation participant specification uses two role relations – departure point' and destination point' which are researched within Subject domain of routes.

The above specification says that there are two trucks on the way, the first one is from the Ruzhany-Agro farm to the plant, the second from the plant to the Vasilishki farm, for each of them the GPS coordinates taken with vehicle sensors. First track also participates in "Car crash" and "Glaze" situations. Car crash situation specification is shown on Listing 8.

Car accident
=> *goods loss**:
3 liters
=> *delay**:
2 hours 30 minutes
=> *accident**:
...
=> *closest place**:
Ruzhany
=> *consequence**:
• *truck repair*
• *driver treatment*
=> *material damage**:
• *300 BYN*
<= *cause-effect**:
• *Glaze*
∋ *participant'*:
Truck 1
∈ *car crash*

Listing 8. Example of car accident situation specification

The knowledge base fragment, shown in Listing 8, indicates that the vehicle was involved in an accident near Ruzhany due to the icy conditions. Because of this accident, 3 liters of transported milk were lost, and the delivery of the remaining milk was delayed by 2.5 hours; car repair and driver's treatment are required. The total damage from the accident is 300 Belarusian rubles.

To visualize the logistics situation on the geographic map, a demonstration prototype of the corresponding user interface component was developed. The map is provided by open geoinformation web service OpenStreetMap [20]. Trucks on the map are placed in accordance with the latest GPS-coordinates fixed in the system (they are related by the ratio of the coordinates* to the corresponding vehicle). The damaged truck is highlighted in color. Logistics situation described in listings 7 and 8 is represented as a map in Fig. 6.

D. Intellectual integration of robots into production complexes

Robotics integration into industrial plants from year to year is increasing, and it requires a reduction of labor costs for the design, development and installation of robotic cells. The robot is an universal machine, but it still requires special preparation of the environment for its work, setting up of the software and algorithm for the certain tasks according to its placement Thus, the universality of the robot as a hardware device is decreased by the specialization and uniqueness of the software that manages it in a particular task. Classical programming of industrial robots based on generation control system for end of tool needs a lot of time and labor resources. Existing systems of off line programming allow to create 3D models of the production line, load robot models and design a robot control algorithm in a virtual environment, and then transfer the control system to a real robot. This approach make possibility to reduce the time and complexity of introducing robotic cells into the production process. However, such a solution is only an automation of the problem of designing and programming an industrial robot. The problem of universal control algorithms development in these software products is not supported, although a significant contribution is the possibility of programming in high-level languages (Java, Python) in contrast to the platform-dependent languages of industrial robots.

Like authors noted before, one of the key production tasks is the task of logistics.

Let us consider an example of a specific production problem in field of logistics solved at the batch manufacturing enterprise with the use of robots in the context of the production of "Khutorok" cottage cheese – finished products packaging.

The using of robots in the task of production logistics is dictated by the high productivity of machines, the need for continuous and accurate processing of goods. The integration of a robot into such a process requires setup of the control system for a specific product, line parameters, etc., which reduces flexibility and production possibilities. The Intelligent industrial robot control system that can be independently reconfigured depending on the type of product, line parameters and production process gives the required universal to the industrial robot cell.

The second class of tasks are organization of few working lines equipped with several robots (see Fig. 7), which interact with each other by processing products going along the line. So on the line for the production of cottage cheese is meant the use of up to 6 robots of various kinematic schemes.

As it was said before, within the proposed approach each robot (or robot complex) is treated as an agent over a shared semantic memory, reacting to events occurring in this memory and specifying all its actions in it.

We can say that the mechanical part of the robot acts as a hardware interpreter of a certain class of programs, and the program part of the robot (programmable controller, etc.) – as a compiler of the robot program stored in semantic memory, into a set of signals understandable by the mechanical part of the robot.

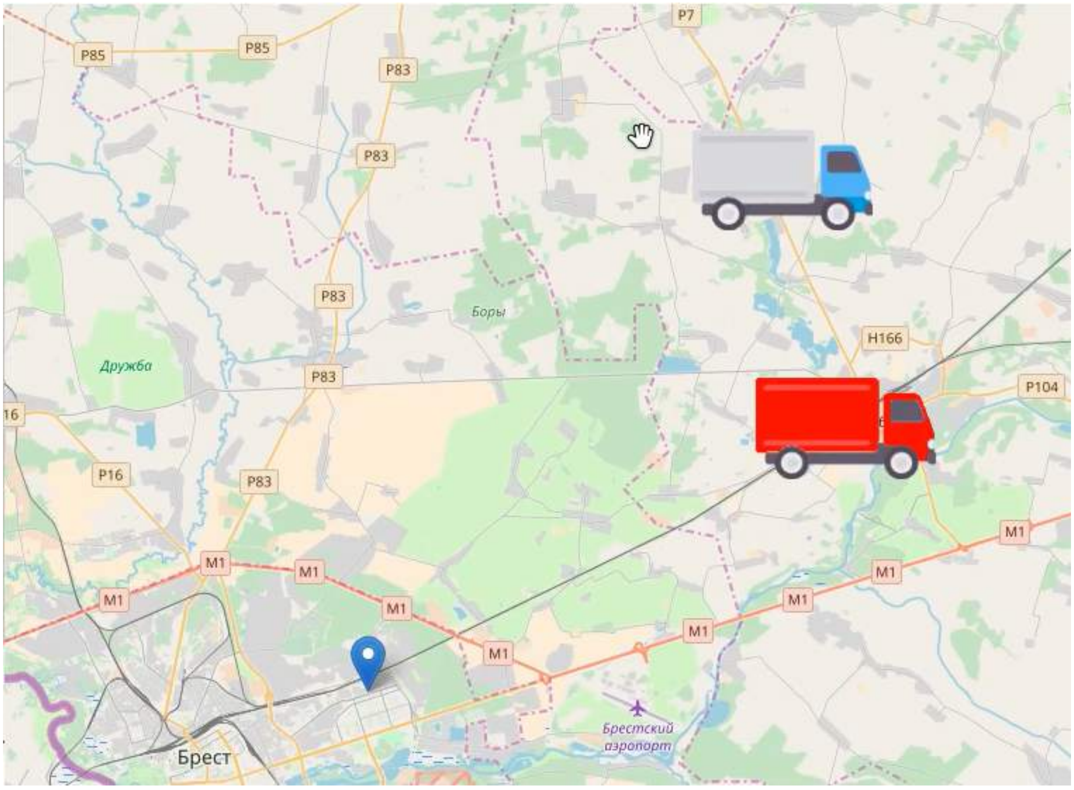

Figure 6. Image of the logistics situation on the map

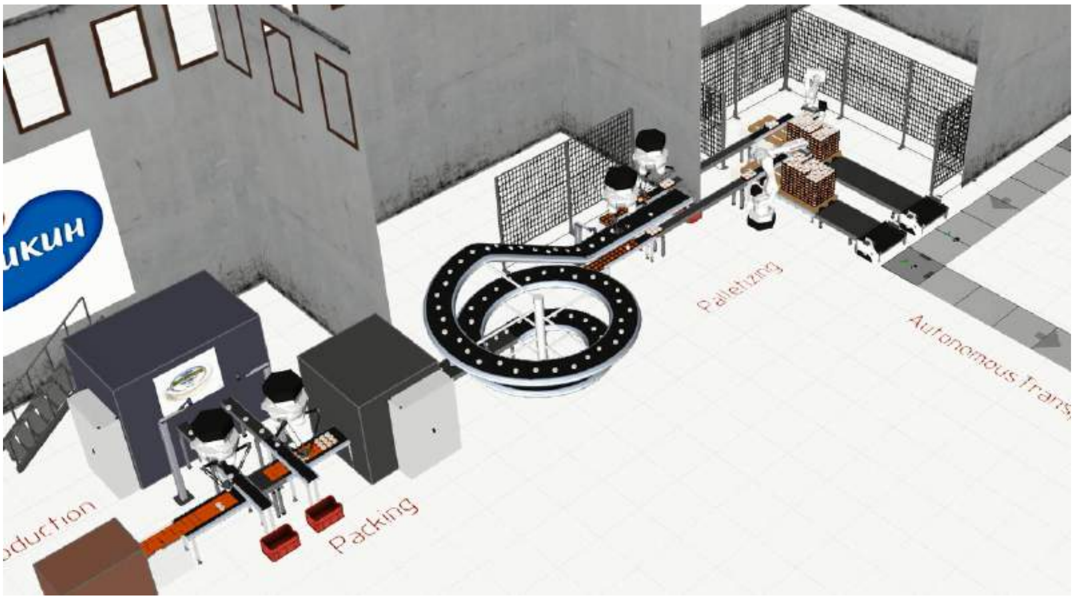

Figure 7. Simulation model of the curd shop with 6 robots

In this case, the addition of a new robot or robot complex is reduced to:

- the development of an ontology of actions that the robot or robot complex is able to interpret, i.e. the description of the denotational semantics of the programs they interpret. In most cases, only classes of actions will differ, other formal means that specify, for example, the sequence of actions or the arguments (operands) of these actions will stay the same.
- the development of tools that allow to transform the

actions specified in semantic memory into signals understandable to the robot.
- the development of the robot activity program itself for the solution of the current class of tasks using the specified ontology of actions.

This approach has the following advantages:
- the robot programming with this approach is clearly divided into logical level (level of performed actions) and hardware level (the level of commands or signals intelligible directly to the robot). This fact provides the following advantages:
  - programming of the robot to perform a specific production task is carried out at the level of operations that are understandable to a specialist in the field of production and does not require special knowledge about the robot's structure, its internal commands and the languages on which programs are compiled at the controller directly;
  - the complexity of the robot reprogramming significantly reduces since reprogramming comes to changing the specification of some actions in the semantic memory and does not require the introduction of changes to a lower level; because the robot program becomes understandable for the automation system itself, i.e. control of the robot's actions and its reprogramming can be carried out in automatic mode, i.e. to be regulated by the system itself;
  - A program of robot actions stored in semantic memory can be visualized in numerous ways, including, in user-friendly graphical languages, which further simplifies the process of manual reprogramming of the robot.
- the presence of shared memory ensures simultaneous consideration of all available robots as a single complex, coordinate their work depending on the needs of the production process, distribute the tasks they solve without having to interact with specific robots in places of physical location. Thus, the management of production processes, monitoring of their implementation can be carried out centrally and remotely;
- as was said before, the approach to communication of system components by means of shared memory provides the flexibility of the system, i.e. allows you to gradually replace manual labor with automatic or introduce more advanced versions of automatic systems without making any changes to the basic automation system. In addition, access to various knowledge stored in the knowledge base allows robots to independently make certain decisions that consider the product nomenclature, their characteristics, knowledge of the product types and ways of products stacking, the location of specific batches, etc.;
- in addition, the specification in the knowledge base of all robot actions provides the possibility of self-learning of the robot based on its own activity, the use of accumulated knowledge in solving typical problems, optimizing its own activity.

Thus, the proposed approach to the intellectualization of production robotic complexes allows us to build a flexible self-adjusting system, which increases the utilization of the robot, shortens the payback period.

III. Conclusion

The article considers the development of the approach to ontology-based design of control systems for batch manufacturing enterprise considering the principles formulated within the Industry 4.0. In addition, the formalization of the ISA88 standard was started in the [10] in terms of the specification in the form of a family of ontologies of syntax and semantics of the graphical language describing the procedural model of the enterprise PFC.

Key points of this paper:

To further increase the manufacturing automation level, it is necessary to consider them from Industry 4.0 — as distributed complexes of control systems, devices, people and services, covering not only production shops, but also warehouses, and interaction with raw material suppliers, wholesale customers and much more.

Considering an enterprise as a multi-agent system over a shared knowledge base built using ***OSTIS technology*** is fully in line with the main principles of the Industry 4.0 concept: interaction, virtualization, decentralization, analysis and response, service orientation, modularity.

A similar approach can be used to organize the interaction of industrial robots within production robotic complexes, which will simplify adding new robots or changing their composition and functionality even to the newbie in the hardware-software platform of a certain robot.

Within Industry 4.0, much attention is paid to human-machine interaction in production automation systems, especially the visual one, which is reflected in the related visual computation concept of Visual Computing [21].

Therefore, the work on formalization of syntax and semantics of the graphic language of the Procedural Function Chart, aimed at the formation of a unified approach to the design of user interfaces of automation systems, also get in the scope of Industry 4.0 direction.

Acknowledgment

The authors would like to thank the staff and administration of the JSC "Savushkin Product" for their support of the work.

References

[1] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, *Recommendations for Implementing the Strategic Initiative Industrie 4.0: Securing the Future of German Manufacturing Industry*, ser. Final Report of the Industrie 4.0 Working Group, 2013.

[2] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *InSystem Sciences (HICSS), 2016 49th Hawaii International Conference on 2016 Jan 5. IEEE.*, 2016, pp. 3928–3937.

[3] *Applied Cyber-Physical Systems. In S.C.Suh et al. (Eds.).* Heidelberg: Springer-Verlag, 2014.

[4] V. Tarasov, “Strategicheskii inzhiniring predpriyatii budushchego: massovoe sotrudnichestvo, internet veshchei, initsiativa «industriya 4.0», chto dal’she? [Strategic engineering of the enterprises of the future: mass cooperation, internet of things, "industry 4.0" initiative, what’s next?],” in *Inzhiniring predpriyatii i upravlenie znaniyami (IP&UZ-2016): sb. nauch. tr. XIX nauch.-prakt. konf. Moskva, FGBOU VO «REU im. G.V. Plekhanova»*, 2016, pp. 57–68, (in Russian).

[5] Y. Tel’nov, A. Danilov, and V. Kazakov, “Setevaya model’ sotrudnichestva i kooperatsii predpriyatii [Network model of enterprises colaboration and cooperation],” in *Inzhiniring predpriyatii i upravlenie znaniyami (IP&UZ-2016): sb. nauch. tr. XIX nauch.-prakt. konf. Moskva, FGBOU VO «REU im. G.V. Plekhanova»*, 2016, pp. 68–72, (in Russian).

[6] V. Tarasov, “Intellektual’nye sredy: tekhnologicheskaya osnova novoi ekonomiki v smart-gorodakh [Intelligent environments: the technological basis for a new economy in smart cities],” in *Reinzhiniring biznes-protsessov na osnove sovremennykh informatsionnykh tekhnologii. Sistemy upravleniya znaniyami. Sbornik nauchnykh trudov XV-i nauchno-prakticheskoi konferentsii (RBP-SUZ-2012, Moskva, MESI, 26-27 aprelya 2012 g.)*, 2012, pp. 180–194, (in Russian).

[7] V. Kupriyanovskii, D. Namiot, V. Drozhzhinov, Y. Kupriyanovskaya, and M. Ivanov, “Internet veshchei na promyshlennykh predpriyatiyakh [Internet of things in industrial enterprises],” *International Journal of Open Information Technologies*, vol. 4, no. 12, pp. 64–67, 2016, (in Russian).

[8] V. Kupriyanovskii, D. Namiot, and S. Sinyagov, “Kiber-fizicheskie sistemy kak osnova tsifrovoi ekonomiki [Cyber-physical systems as the basis of the digital economy],” *International Journal of Open Information Technologies*, vol. 4, no. 2, pp. 19–25, 2016, (in Russian).

[9] (2017, Nov.) Garbrecht S. The three rules of industrial Operations Management and Industrial IoT Applications. [Online]. Available: https://www.linkedin.com/pulse/three-rules-industrial-operations-management-iot-steven-garbrecht/

[10] V. Taberko, D. Ivanyuk, V. Golenkov, K. Rusetskii, D. Shunkevich, I. Davydenko, V. Zakharov, V. Ivashenko, and D. Koronchik, “Proektirovanie predpriyatii retsepturnogo proizvodstva na osnove ontologii [Designing enterprises of batch production on the basis of ontologies],” *Ontologiya proektirovaniya [Ontology of design]*, no. 2, pp. 123–144, 2017, (in Russian).

[11] V. Tarasov, *Ot mnogoagentnykh sistem k intellektual’nym organizatsiyam [From multi-agent systems to intelligent organizations].* M.: Editorial URSS, 2002, (in Russian).

[12] (2017, Nov.) ISA88, Batch Control. [Online]. Available: https://www.isa.org/isa88

[13] T. Gavrilova, D. Kudryavtsev, and D. Muromtsev, *Inzheneriya znanii. Modeli i metody: Uchebnik [Knowledge engineering. Models and methods: Textbook].* SPb.: Izdatel’stvo «Lan’», 2016, (in Russian).

[14] B. Shvedin, *Ontologiya predpriyatiya: ekspirientologicheskii podkhod: Tekhnologiya postroeniya ontologicheskoi modeli predpriyatiya [Ontology of the enterprise: the experimental approach: The technology of constructing the ontological model of the enterprise].* M: Lenand, 2010, (in Russian).

[15] D. Pospelov, *Situatsionnoe upravlenie. Teoriya i praktika [Situational management. Theory and practice].* M.: Nauka, 1986, (in Russian).

[16] M. Robson and P. Ullah, *A practical guide to business process reengineering.* Gower Publishing, Ltd., 1996.

[17] *Instrumentation, Systems, and Automation Society, 2001, ANSI/ISA-88.02-2001 Batch Control Part 2: Data Structures and Guidelines for Languages.* ISA, Research Triangle Park, USA, 2011.

[18] D. V. Shunkevich, “Ontology-based design of knowledge processing machines,” pp. 73–94, 2017.

[19] Y. Kolchmakhin, “Skladskie biznes-protsessy [Warehouse business processes],” *Skladskoi kompleks*, no. 1, pp. 64–67, 2014, (in Russian).

[20] (2017, Nov.) OpenStreetMap. [Online]. Available: https://www.openstreetmap.org/

[21] A. Stork, “Visual computing challenges of advanced manufacturing and industrie 4.0,” *IEEE computer graphics and applications*, vol. 35, no. 2, pp. 21–25, 2015.

ПРОЕКТИРОВАНИЕ ПРЕДПРИЯТИЯ РЕЦЕПТУРНОГО ПРОИЗВОДСТВА В КОНТЕКСТЕ НАПРАВЛЕНИЯ INDUSTRY 4.0

Таберко В.В., Иванюк Д.С., Касьяник В.В. (ОАО «Савушкин продукт», г. Брест, Республика Беларусь)

Головко В.А. (Брестский государственный технический университет, г. Брест, Республика Беларусь)

Гулякина Н.А., Русецкий К.В., Шункевич Д.В., Борискин А.С., Гракова Н.В. (Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь)

В данной работе предлагается развитие онтологического подхода к проектированию предприятий рецептурного производства, заключающееся в переходе от рассмотрения производственных процессов в рамках одного предприятия, к рассмотрению процессов, охватывающих, в соответствии с концепцией Industry 4.0, и смежные предприятия – поставщиков сырья (молочные фермы) и оптовых потребителей продукции (магазины, торговые сети). Особое внимание уделяется логистическим процессам – приводится краткое описание основных логистических процессов, касающихся производства творога, структура предметной области логистики и пример формализации нештатной логистической ситуации. Обосновывается соответствие многоагентной системы предприятия со взаимодействием агентов через общую память основным принципам Industry 4.0. В рамках формализации стандартов рассматривается формализация внешнего языка спецификации процедурных моделей Procedure Function Chart, определенного во второй части стандарта ISA-88. Формализация внешнего языка позволяет организовать взаимодействие с пользователями системы автоматизации предприятия на основе унифицированного подхода с использованием понятного инженерному персоналу языка диаграмм. Рассматривается также агентно-ориентированный подход к организации взаимодействия роботов в рамках роботизированных производственных комплексов, основанный на взаимодействии через общую семантическую память.