



Semantic Reference Model for Individualization of Information Processes in IoT Heterogeneous Environment

Dmitry Mouromtsev

Article



Citation: Mouromtsev, D. Semantic Reference Model for Individualization of Information Processes in IoT Heterogeneous Environment. *Electronics* **2021**, *10*, 2523. https://doi.org/10.3390/ electronics10202523

Academic Editors: Dmitry Korzun, Ilya Shegelman and Anatoly Voronin

Received: 30 August 2021 Accepted: 13 October 2021 Published: 16 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. 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 (https:// creativecommons.org/licenses/by/ 4.0/). Faculty of Software Engineering and Computer Systems, ITMO University, 197101 St. Petersburg, Russia; mouromtsev@itmo.ru

Abstract: The individualization of information processes based on artificial intelligence (AI), especially in the context of industrial tasks, requires new, hybrid approaches to process modeling that take into account the novel methods and technologies both in the field of semantic representation of knowledge and machine learning. The combination of both AI techniques imposes several requirements and restrictions on the types of data and object properties and the structure of ontologies for data and knowledge representation about processes. The conceptual reference model for effective individualization of information processes (IIP CRM) proposed in this work considers these requirements and restrictions. This model is based on such well-known standard upper ontologies as BFO, GFO and MASON. Evaluation of the proposed model is done on a practical use case in the field of precise agriculture where IoT-enabled processes are widely used. It is shown that IIP CRM allows the construction of a knowledge graph about processes that are surrounded by unstructured data in soft and heterogeneous domains. CRM also provides the ability to answer specific questions in the domain using queries written with the CRM vocabulary, which makes it easier to develop applications based on knowledge graphs.

Keywords: semantic reference model; knowledge graphs; information processes individualization; IoT process

1. Introduction

These days ontology modeling has become the standard de facto for knowledge and data representation, especially for applications in complex domains, where unstructured or semistructured data are the main source of information. The process modeling in IoT heterogeneous environment that includes a mess of connected devices and informal user needs is one of these domains. The dynamic nature of processes imposes requirements to the methods of information modeling that should include a certain ability to represent the sequences of events in time on the one hand and the domain-specific conceptualization on the other hand. Despite the expressivity of ontology engineering, its potential has not been fully revealed for process modeling, where methods without formal semantics, such as UML or BPMN, remain very popular.

The problem of process modeling for IoT is becoming topical as the number of connected devices is growing and the level of their intelligence is constantly increasing. Widely used process models are intended for tasks such as process design and refactoring, process observation, and explanation of process rationale. Herewith, the existing models are developed to describe some templates for a typical process from different perspectives, such as Activity-oriented, Product-oriented, Decision-oriented and other perspectives. Unfortunately, one of the most important requirements these days has not been properly addressed, namely the ability to individualize processes to meet the changing user needs.

The problem of individualization is not only the syntheses of the individual trajectory of the process, but the creation of such an individualization model, which together with the domain ontology and the process model will generate the required number of options with the possibility of their adaptation under the changing process conditions. Traditional approaches to business processes modeling consider mainly process design and do not take into account the dynamic nature of IoT processes and do not meet the requirements of individualization of these processes. In [1,2] Configurable Process Models (CPMs) are discussed, which are aimed to provide flexibility and systematic reuse of process models and avoid the development of processes from scratch.

In [1], a formal definition for configurable processes as the inverse of inheritance of dynamic behavior is proposed. It allows configurations to restrict the process model instead of adding new functionality in configurable process modeling languages. This makes it possible to extend such language as Event-driven Process Chains (EPCs) and support a postponement decision-making based on information available at the processes run-time.

In [2], the focus is made on process-aware information systems (PAIS) and IoT resource management by means of configurable IoT allocation operators for "modeling IoT resource variability at the CPM level". Modeling of IoT specific features, properties and behavior, such as Replication and Shareability, at the CPM level allows consideration of the preexisting knowledge about similar IoT resource specifications while reusing and redesigning process models.

In other words, CPM-based models allow the explicit capture explicitly and semantic inheritance of run-time process management. However, these models are still rigid with modeling user needs and IoT environment. To overcome this limitation, the modeling approach must explicitly allow working with data and knowledge models that describe business requirements and relevant constraints in the user language, instead of IoT specification application. The problem of existing approaches to ontological modeling of processes remains the conceptualization of static predefined objects involved in the process. Considering processes realization, individualization, perception and explanation of the current status of the process are important factors.

Typical IoT systems are sensors and actuators, tags, readers and collectors, which form the IoT environment. However, this environment does not always satisfy the real vision of the user. For example, the control of light in a room actually depends on the level of lighting, which in turn depends on the parameters of the luminaires, brightness of natural light and blackout windows. In the aggregate, these processes can include the work of dozens of IoT devices, the user knows nothing about. Such a reasoning is true for any IoT-enabled process. Thus, the core of the individualized process-aware systems architecture should be a knowledge base of user needs and the subject area. To build ontological knowledge base, the so-called "upper ontologies" are required. A set of upper ontologies covering different aspects of a specific problem or task compose a conceptual reference model.

OASIS SOA Technical Committee defines a reference model as "an abstract framework for understanding significant relationships among the entities of some environment [...]. A reference model is based on a small number of unifying concepts [...]. A reference model [...] does seek to provide a common semantics that can be used unambiguously across and between different implementations." [3]. In other words, such model is a domain-specific ontology that facilitates communication and development processes. The core of a reference model is a conceptual model consisting of a set of concepts which allow people know, understand, or simulate a subject, the represented system and its environment. Therefore, the construction of a conceptual model requires conceptualization or generalization of the process [4].

When we consider a certain application domain for a reference model as a set of processes described by its purpose and the associated outcomes, we should refer to some Process Reference Model (PRM). PRM typically provides a generic functionality and can be used more than once in different models. Thus, a starting point to create a process model for a specific purpose is the reuse and customization of a PRM according to a supporting methodology instead of model creation from scratch [5].

The key benefits of reference process models are timesaving while designing new models, following industry best-practices, and adaptation of the developed models to the

individual needs of a user [6]. Reference models may be constructed both "top-down" and "bottom-up" [7]. The first approach follows generally accepted theories and principles. The second approach is based on the analyses and application of real-world data, classification and information retrieval techniques. In [8], a new approach to inductive reference model development is proposed. This method is based on an execution-semantic similarity measure and capturing the commonalities of the input models in a behavioral profile. As a result, it becomes possible to derive a reference model subsuming the input models' semantics instead of their structure and reach a higher level of flexibility and applicability to inductive reference modeling.

The final and the most important feature of the considered reference models is a semantic layer that provides capabilities for reasoning and knowledge base development aimed to support individualized purposes. Recent trends here are to build the so-called knowledge graphs (KGs), which include graph repositories of semantic metadata (or knowledge formalized using special formal semantic languages such as RDF) and ontologies, a semistructured conceptual scheme for process data. The latter quality distinguishes knowledge graphs from databases, making it possible to perform intelligent data analyses by means of linking, reuse, and enrichment operations in the data management lifecycle.

Summarizing the above, it is necessary to highlight some requirements for the conceptual reference model that need to be considered to build an effective lifecycle of individualized IoT processes support:

- development of a semantic reference model of process individualization will allows the building of specific knowledge graphs for more flexible and adaptive information process management in IoT;
- development of a proper CRM logical scheme to automatically link the levels of data representation in IoT systems and the levels of description of user needs considering their business requirements expressed in domain-specific terms;
- consideration of possible ways to design a representation model with lower scheme complexity, which facilitate the creation and use of a hybrid approach to build KGs, including several techniques such as KGs Embeddings or Graph Neural Network;
- solving the problem of computational complexity in process-aware information systems to apply hybrid AI methods that include KGs;
- identification and description of patterns for data representation and query that avoid excessive domain-specific constructs while designing information models. The information about individual needs that is used for model development typically exists in an informal form. The datasets typically contain data generated by multiple subjects, although for an individual the collected data may be relatively small.

2. Related Works

2.1. Reference Models

A comprehensive analysis of the concept of the reference model and approaches to its construction and its role in information modeling is provided in [9]. Ahlemann in his work considers several models starting from one of the first reference information models for project management in the architecture, engineering, and construction (AEC) industry published by Froese, who called it a "standard model" [10] and later developments such as a combined reference information model for process and project controlling published by Schlagheck [11].

2.2. Process Models

The review of the basic approaches for a modeling if information processes are described in [12–14]:

- Descriptive models such as EPC, IDEF, UML and BPMN. These models lack formal semantics necessary to implement the inference mechanism.
- Procedural languages such as BPEL and XPDL. These languages are not intuitive enough for human use, there is no declarative rules for the reasoning mechanism.

- Object-oriented programming such as BORM, PML, BOOPM. These models are mostly used in software development tasks and not intended to be modeling languages for information processes.
- Formal languages such as PSL, Pi-Calculus, Petri-Nets are based on strict mathematical principles and difficult to understand for domain experts without additional in-depth training.
- Hybrid models based on neural networks such as Transition-Based Neural Model, Narrative Event Evolutionary Graph Construction (NEEG). These models are computationally effective but lack explanatory features.
- Ontological models such as The Business Process Modeling Ontology and BPAL (Business Process Abstract Language.) provide a good comprehension level and combined with hybrid neural models give a lot of possibilities.

2.3. Ontologies, Knowledge Graphs and Hybrid AI

The role of ontologies in the process modeling is to provide a vocabulary for semantic annotations for real-world data from heterogeneous sources or metadata, which can be represented through connected ontology concepts. These concepts are linked by object properties and annotated by data properties that are facts about objects or processes. Being populated with a large number of facts and deployed as a service, ontologies become knowledge graphs, or information resources accessible both for software agents via API and for humans via user interface. It is possible to use knowledge graphs as structured and annotated knowledge base via endpoints and semantic queries. A detailed review of the use of ontological approaches is given in [15].

A good example of open and popular KGs is Wikidata (https://www.wikidata.org/ wiki/Wikidata:Main_Page, accessed on 14 October 2021) where we can find several useful modeling patterns data representation techniques. so-called reification, where a set of necessary values is encapsulated in a statement or a container, is a flexible approach to model entities for which specific values of their attributes are determined by the context. A discussion of various options for implementing reification in knowledge graphs is given in [16]. Formally, reification fits with higher-order logics, which allows statements to be made about other statements in one graph.

Consider how to combine process-specific data, relevant metadata and referenced sources in an RDF graph for the following statement "the product output of the process <processID> is 123456. The process was carried out by <personID> on 2021-08-30Z according to <chartID> with the parameters set parametersID>":

<processID> a rdf:Statement;

| rdf:subject | ex:Process; |
|---------------|--------------------------------|
| rdf:predicate | ex:product_output; |
| rdf:object | 123456; |
| ex:operator | <personid>;</personid> |
| xsd:date | 2021-08-30Z; |
| ex:flow_chart | <chartid>;</chartid> |
| ex:parameters | <parametersid>.</parametersid> |

This graph allows the answering of such questions as "Who carried out the process and how?", "When was it done?" and so on. The predicate ex:product_output can be enriched with necessary additional attributes and this provides both provenance and trust in the model.

However, the details of the process statement are attached to <processID> rather than the ex:product_output predicate and the reification semantics in the RDF standard does not explicitly declare the statement "The product output of <processID> is 123456" as a triple:

<processID> ex:product_output 123456.

At first glance, reification seems to be a convenient way to represent individualized processes, since it naturally supports versioning, flexibility, and extensibility of the model. However, in terms of new hybrid AI approaches, it is also necessary to consider the capabilities of representative models to ensure the performance of methods such as graph embeddings and tensor factorization, Graph Neural Networks (GNN), Graph Deep Learning (GDL), Graph transformers and so on. Hybrid AI includes new approaches that allow a sophisticated modeling of complex processes based on combinations of graph models and machine learning techniques. Therefore, the excessive complexity of knowledge graphs reduces the efficiency of machine learning methods. In References [16,17], an analysis of the performance of different types of reification knowledge graphs is presented. Additionally, in [16] it is shown that the use of reification can cause the generation of auxiliary statements that are two orders of magnitude larger than the original number of triplets. This in turn consumes a lot of memory and requires optimization.

3. Individualization Concept in IoT

3.1. The Problem of Individualization in IoT Processes

The concept of individualization is mainly used in the context of product development and e-commerce. In this sense the role of IoT is to facilitate enabling related information services or deliver process simulation is performed using the operator customized and adopted content. The work [18] proposes an evaluation of several kinds of IoT services that foster product service systems. At the same time, challenges such as integrating IoT technologies into the production process that enable efficient customization of the process to individual needs and environment remain largely unresolved. The marketing point of view of the process individualization is also related to the problem of automated data analyses which assumes levels of data integration and semantic interoperability in the system. In [19] authors declare that individualized approaches require a data-driven strategy. Individualization is considered to be an approach that acknowledges the needs of the individual as attributes that make him unique. In [20] they go even further and define a hyper-individualization strategy that "leverages artificial intelligence and real-time data to help understand true intent behind the behavior of individual users".

Individualization also may be treated as mentioned Section 1 configurable IoT-aware processes based on concrete business requirements to some extent. In [2] modeling of IoT resource variability at the CPM level is performed using three main operators: Configurable IoT Assignment operator, Configurable IoT Replication operator, and Configurable IoT Shareability operator. The introduction of these operators solved the problem of meeting business requirements by individualization of IoT-aware CPMs into a specific process variant via the control-flow and IoT resource perspective.

In [1] the notion of Labeled Transition Systems (LTS) is proposed as a generic model integrating possible process configurations' variants into one model. This approach treats individual models as the inheritance "the least common multiple" of all superclasses of all possible process configurations. A configured process model obtained in this manner may probably not satisfy all individual requirements but should guide the user to a satisfactory decision made at run-time.

Summarizing the above, the process individualization can be defined as the construction of a model of the needs and behavior of the individual in the context of a variety of alternatives for the process development. The main purpose of this model is the integration and intelligent analysis of data for decision-making on the maintenance and modification of the process to achieve greater efficiency. Additionally, the model should provide the domain expert with templates for data representation at the abstract level, as well as a facility for semantic queries to the data.

3.2. A Process Individualization from Hybrid AI Perspective

When constructing an individualized process, it is generally difficult to specify the sequence of events in the process, because there may be any number of options for making

the event combinations, depending on changing conditions and the needs of the individual. Traditional AI approaches can help to solve the problem of classifying events or recognizing certain patterns or anomalies in processes. However, this is not enough to solve the individualization problem, since the decision-making is left to the expert or the user, who must inevitably be involved in the data analysis loop. New hybrid AI methods such as GNN, GDL or graph transformers can be used to automate the synthesis of individualized process trajectories [21]. These methods exploit architectures that use a pipeline consisting of both latent methods such as embeddings, recurrent neural networks, forward propagation networks, etc., as well as attention mechanisms, representational models, and so on. The role of data representation models is configuration of the parameters for use of hybrid AI methods, as well as interpretation of their results.

In the context of hybrid methods, the problem of process individualization can be reduced to the problem of transforming data in the form of a graph at the input into a sequence of the most relevant process events at the output. Graph-to-Sequence models are efficient in tasks such as machine translation or natural language generation and use Abstract Meaning Representation as the input graph [22]. A significant advantage is the fact that the implementation of the Neural Graph-to-Sequence Model allows the avoidance of explicit coding of the sequence of events [23]. To apply Graph-to-Sequence approaches to process modeling, it is necessary to construct a process knowledge graph that uses more complex axioms of OWL semantics, describing changing user needs and environment observation, in addition to node labels and links.

In the remaining of this paper, a new top-level ontology for individualized process is presented. We also show how the proposed ontology can be applied as a reference model for the data representation level for the creation of a knowledge graph in a weakly formalized domain. We speculate this can be beneficial in tasks such as process individualization and leave further investigation and optimization of related hybrid AI architecture to future work.

3.3. Motivation Example

As an example of the complex task of individualization of IoT-enabled processes, consider the subject area of a precise agriculture. The technology of high-precision (precise or coordinate) agriculture makes it possible to take into account the peculiarities of existing heterogeneities of relief, soil-forming structures, the impact of ground and rainwater, the impact of wildlife and human activity itself. The precise agriculture is an integrated high-tech system of agricultural management, based on global positioning data, geoinformation systems, yield monitoring technologies, variable rate technology and remote sensing technologies, providing objective and timely updated information on land use and crop condition. Together, all these systems form a complex heterogeneous IoT environment with multiple data sources in different formats. The precise agriculture is based on the so-called agro-technology, which includes a sequence of typical operations. Each of these operations is implemented as a set of steps, the sequence and parameters of which depend significantly on the individual field state and environment features. In certain critical situations, some of these steps may be canceled or, on the contrary, new steps may be added to compensate for the negative impacts and scenarios of the crop growth process.

Figure 1 shows a general diagram of precise agriculture processes. At the same time, despite the high level of automation, the decision-making process for managing these processes is a black box in terms of the system and is completely controlled by humans.

The objective of management is the production and delivery to warehouses of specified types of products in specified quantities with minimum resource consumption and losses. Risks in agriculture may reach up to 40% of the lost revenues. At the moment, the methodology for achieving the objective is based on experience of applying the same cycle of production for many years. It consists of sequential actions of operators to implement each process in the order specified by the agro-technology with discrete reconciliation and adjustment of the state of planned and actual indicators before the start of each process and after its completion.



Figure 1. Simplified diagram of precise agriculture processes.

The challenge for the next-generation precise agriculture management system is to create an integrated data repository for processes, as well as a recommender system for shaping individual processes for each individual field. At the same time, such recommendations should take into account not only agrophysical, chemical and other characteristics of the soil, but also dynamically changing weather conditions, spontaneous effects of wildlife, pests, etc., as well as possible changes in the objectives of the producer. The proposed conceptual reference model discussed below is supposed to be the upper-level ontology for the precise agriculture knowledge graph and the process control system.

4. Formal Framework for the Reference Model

4.1. The Ontological Categories of Individualized Process

Traditional approaches to modeling processes do not imply different ways of representing spatio-temporal aspects for the processes themselves and the objects involved in them. Such approaches provide the simplicity and homogeneity of the models and the relative simplicity of their construction. However, this unification makes it impossible to separate the reasoning into different levels, in our case, the levels of the individual's needs and the means of satisfying them, and the level of development of a process over time. In some sense, the absence of such levels can be compared to trying to speak a language that has only verbs but no nouns.

The work [24] describes an ontological view of the problem of processes and objects. In particular, Heller and Herre in their work declare the difference between endurants (or "continuants") and processes according to their relation to time. An endurant is an individual that does not have temporal parts or phases and for which time is in a sense a container. Endurants may be indexed by time boundaries. On the contrary, processes have temporal parts and thus cannot be present at a time-boundary. Heller and Herre call the category of processes to which they belong as "occurrents".

A typical process consists of a pre-planned sequence of steps. However, this is not possible in a situation where the possibility or impossibility of performing each step depends not only on the previous one, but also on changing external factors and conditions. These can be such factors as available resources at any given time, environmental conditions (e.g., weather), connectivity problems, and most importantly, the changing needs of the individual. In accordance with this, we can say that the entire process is divided into three parts: those steps that have already been implemented and are unchangeable. This is the history of the process or its tail. The current moment in time and operations in progress, observations of conditions, refinement of goals. In addition, a multitude of future alternatives. Only depending on the available data in the current setup, it is possible to plan a single next step or a sequence of several dependent steps. The parameters of the next steps of the process are shaped as a translation of the user's needs and goals in the context of current observations and resources to a set of upcoming actions, taking into account the path already taken. In other words, the jump to the following state forms a path in the KG of the domain as some projection, which in turn is compared to the tail and the states that can potentially form the continuation of the process. Thus, the reference model should represent the dependencies between the steps of the process, the individual's goals (global and local, which may conflict), resources, and observations of conditions on two levels of abstraction.

The most similar approach to the modeling described here was previously developed for Basic Formal Ontology [25]. In particular, to resolve the collision of the representation of the process as a sequence of pre-planned events and the changing conditions and requirements of the individual influencing this process, the idea of the dichotomy between continuant and occurrent ontologies may be taken as a basis. This dichotomy as a part derived from Zemach [26] is implemented in the BFO 2 [27].

4.2. The Reference Model Specification

The semantic reference model for individualization of information processes was developed based on several well-known upper ontologies. Mainly this model may be considered to be an extension of mentioned above BFO 2 model. However, also, there are several other ontologies were reused. They are General Formal Ontology (GFO) [28] and MAnufacturing's Semantics Ontology (MASON) [29]. To model more specific entities of real-world processes and scenarios The Dublin Core[™] Metadata Initiative (DCMI) Metadata Terms [30] and The Suggested Upper Merged Ontology (SUMO) vocabulary [31] are widely used. The SNAP and SPAN [32] approach is taken as the methodological basis for modeling the dynamic nature of information processes. In particular, this modeling methodology consider processuals as happenings which involve participants of some kind, and they are dependent on their participants and the relations obtaining between them. SNAP and SPAN theoretical framework allow the representation of processes as wholes of self-connected extended processual entities.

For simplicity of the developed reference model, *owl-import* was not applied for the BFO and other ontologies as well as the *EquivalentClass* statement. Instead of this, where it is necessary, the corresponding links to the entities of imported ontologies are indicated in the *rdfs: comment* annotation. Although this information has no additional semantic value for model designing, it can be useful for knowledge engineers if there is a desire to refer to the modeling methodology of BFOs, GFOs, etc.

The list of key classes of the semantic reference model for individualization of information processes and mappings to known relevant ontologies is given in Table 1. The domain name IIP–Individualization of Information Process is used for the original developed classes and properties to distinguish them from related entities of existing ontologies (BFO, GFO, etc.).

The IIP:Process class is a template or prototype class for certain processes. This class is needed for general knowledge representation about a particular process. The IIP:Process Implementation, IIP:Process Implementation Tail and IIP:Process Implementation Head classes are used to represent data about a particular process implementation at any point in time. For a more detailed description of the atomic steps of a process the IIP:Step of Process is used. The IIP:Object of Process class is for data about real-world objects that are produced or changed during the implementation of a process. Data on changes in the characteristics of these objects is represented with the class IIP:Observation. The three classes IIP:Process Owner, IIP:Process Inspector and IIP:Process Executant are designed to define different roles in process control. The IIP:Modification class is needed to describe explicitly how the properties of process. The IIP:Machine class describes the material resources that are needed Table 1. The list of key classes of the Semantic reference model for individualization of information processes.

| IIP Class | SubClass of | Definition | Mapping to Concepts in Referential Ontologies |
|------------------------------------|-------------------------------|--|--|
| IIP:Process | BFO:process | P is subclass of the process BFO class (BFO:process, BFO2 Reference: [083-003]) and is an occurrent that has temporal proper parts as instances of <i>Step of Process</i> and for some time t <i>depends on</i> some <i>resources</i> . | GFO:Process |
| IIP:Process Implementation Tail | BFO:history | PT is subclass of the history BFO class (BFO:history, BFO2 Reference: [138-001]) and is a continuant being a history of a process that is the sum of the totality completed process steps. | GFO:History |
| IIP:Process Implementation Head | BFO:p-boundary | PH is subclass of the <i>process boundary</i> BFO class (BFO:p_boundary, BFO2 Reference: [084-001]) and is a temporal part of a <i>process</i> as an instance of <i>Step of Process</i> and has no proper temporal parts | GFO:Event |
| IIP:Process Implementation | | Union of Process Implementation Tail and Process Implementation Head. The Process Implementation is a specific concept for time-dependent relations. | |
| IIP:Observation | BFO:process-profile | O is subclass of the <i>process-profile</i> BFO class (BFO: process-profile, BFO2 Reference: [093-002]) and is a selective abstraction that is applicable when measurements are made of changes in single qualities. | GFO:State |
| IIP:Step of Process | BFO:realizable | SP is subclass of the <i>realizable entity</i> BFO (BFO:realizable, BFO2 Reference: [058-002]) and is of a type instances of which are <i>realized</i> in some processes (of a specific type). | GFO:Event |
| IIP:Object of Process | BFO:object | SP is subclass of the object BFO (BFO:object, BFO2 Reference: [024-001]) and is of a type instances of which are developed during some processes (of a specific type). | GFO:Material object |
| IIP:Process Owner | BFO:role | PO is subclass of the <i>role</i> BFO class (BFO:role, BFO2 Reference: [061-001]) and is of a type instances of which <i>implement</i> of some <i>process</i> (of a class BFO:process, BFO2 Reference: [083-003]). PO is responsible for the implementation of the process and the decisions making. | MASON: Handling Operator GFO:Processual role |
| IIP:Process Inspector | BFO:ic-independent continuant | PI is subclass of the <i>independent continuant</i> BFO class (BFO:ic-independent continuant, BFO2 Reference: [017-002]) and is of a type instances of which <i>checks</i> a <i>quality</i> (of a class BFO:quality, BFO2 Reference: [055-001]). | MASON: Procedures expert GFO:Relational role |
| IIP:Process Executant | BFO:role | PE is subclass of the <i>role</i> BFO class (BFO:role, BFO2 Reference: [061-001]) and is of a type instances of which <i>executes</i> some process step. PE is responsible for the execution of the process steps. | MASON: Operator GFO:Processual role |
| IIP:Modification | BFO:function | M is subclass of the <i>function</i> BFO class (BFO:function, BFO2 Reference: [064-001]) and is of a type instances of which updates of some quality. | GFO:Change |
| IIP:Machine | GFO:Material object | M is subclass of the <i>Material object</i> GFO class (BFO:Material_object) and is of a type instances of which are required by some <i>process</i> to be executed. | MASON: Machine resource |
| IIP:Consumables | GFO:Material object | C is subclass of the <i>Material object</i> GFO class (BFO:Material_object) and is of a type instances of which are consumed by some <i>process</i> to be executed. | MASON: Raw material |

The description of key object properties of the semantic reference model for individualization of information processes is given in Table 2. Several relevant properties are not included in the table because they are used with the same semantics as they defined in BFO 2 ontology. For simplicity reasons inverse properties are excluded from the model. Figure 2 shows a diagram with the main IIP classes and object properties.

In the model described here there are not the additional predicates and reification-style classes discussed in Section 2.3. This was done to improve the efficiency of the model in combination with hybrid AI methods. From a practical point of view, the omission of reification ensures that the actual values of predicates can be handled without additional complex aggregation queries. It is also made possible to keep the volume of the graph to a minimum, with no artificial new nodes-statements that do not contain any knowledge.



Figure 2. The main IIP classes and object properties diagram.

Table 2. The list of key object properties of the Semantic reference model for individualization of information processes.

| | | 2 4 14 | |
|-------------------------|--|--|--|
| IIP Property | Equivalent or Sub Property to | Definition | Domain/Range |
| IIP: implements process | BFO: participates in at some time (BFO2 Reference: [086–003]) | an instance-level relation between a process and a process owner at which the process owner implements the process under existing conditions | Process Owner/ Process |
| IIP: is included in | BFO: realized-in (BFO2 Reference: [106–002]) | an instance-level relation between a step of a process and a process at which the step of a process performed as a part of the process under existing conditions | Step of Process/ Process |
| IIP: is checked by | BFO: s-depends-on_st (BFO2 Reference: [083–003]) | an instance-level relation between a quality of some process step and a process inspector which does a quality assessment of a process | Quality/Process Inspector |
| IIP: executes | BFO: has_participant (BFO2 Reference: [086–003]) | an instance-level relation between a process executant and a process step at which process executant executes the ongoing process step | Process Executant/ Step of Process |
| IIP: is performed in | BFO: participates in at some time (BFO2 Reference: [086–003]) | an instance-level relation between a step of process and a process implementation (tail or head) at some specified time <i>t</i> | Step of Process/ Process Implementation |
| IIP: is observation for | BFO: profile-of (BFO2 Reference: [093–002]) | an instance-level relation between an observation of a process and a process at some specified time t | Observation/ Process Implementation |
| IIP: requires | owl: topObjectProperty | an instance-level relation between a step of some process and a machine by means of which a process executant executes the ongoing process step | Step of Process/ Machine |
| IIP: is dependent on | owl: topObjectProperty | an instance-level relation between a step of some process and a consumable resource which the execution of the process step depends on | Step of Process/ Consumables |
| IIP: causes | BFO: has-f_st (BFO2 Reference: [070–001]) | an instance-level relation between a step of some process and a modification of an object which causes a change of its quality | Step of Process/ Modification |
| IIP: affects | owl: topObjectProperty | an instance-level relation between a modification and an object of some process which affects an object's quality | Modification/ Object of Process |
| IIP: is developed in | owl: topObjectProperty | an instance-level relation between an object of some process and the process implementation in which the object is developed | Object of Process/ Process Implementation |
| IIP: has quality | owl: topObjectProperty | an instance-level relation between an object of some process and the object's quality | Object of Process/ Quality |

5. Evaluation and Real-World Example

This section will examine the application of the proposed conceptual reference model to the example of precise agriculture described in Section 3.3. As a specific problem, consider a situation in which an agronomist must decide whether to adjust agrochemical and agrophysical properties of the soil for a particular field, which should allow the minimization of risks in the agricultural process. Today agronomists use a complex IoT-enabled systems for collecting heterogeneous data, but the decision-making process remains informal and is based on the user experience. The various formats of data available include sparse text files, tables, interfaces, and documents collected by the agronomist to analyze each implementation of the process. However, the knowledge bases are not used for decision-making. Therefore, without modeling instruments such as the proposed IIP CRM it is practically impossible to formalize, represent and explain different individual processes related to some particular agricultural subject.

During this evaluation task we apply the developed and described above CRM to the set of domain-specific concepts. These concepts relate to the environment of the process but at the same time they are not an informational model of the process or its vocabulary. Thus, the goal of this evaluation is, on the one hand, to build a knowledge graph that integrates disconnected concepts into a single model, and on the other hand, to demonstrate the possibility of the questions answering with the assistance of the created information resource. An example of such a question will be presented in the form of SPARQL-query.

The experimental knowledge graph of the precise agriculture was designed iteratively. The necessary measurement data, technological flow charts of agro-technologies, reports on production processes, etc. were provided by the Agrophysical Research Institute of St. Petersburg. The information presented includes a data set for several years of wheat production of other crops. For each year, there are results of measurements of various soil properties, as well as reports of technological operations (artificial modifications of soil properties) that were performed during production process. The pipeline of KGs development included the modeling phase in Protégé editor, and a knowledge graph population stage. For the second stage, scripts were developed in Python using the rdflib 6.0.1 library. The examples in this article have been translated into English and adapted for understanding by non-experts in precise agriculture terminology and notations.

At each step of KGs development, the available data and unstructured information that the experts use to make decisions were analyzed. As a result of the analysis, the identified entities were mapped to IIP CRM concepts. The equality links between concepts such as sameAs were not used due to non-trivial semantics of such links, especially in complex domains, and potentially they may raise some problems with regards to inference issues [33]. Instead, the inheritance of a domain-specific classes and adding necessary individuals were done. In some cases, where necessary, multiple inheritance has been used. In this paper, we present an example of the resulting ontology (a fragment). The OWL file can be downloaded here (http://purl.org//iip-srm/ATO-example, accessed on 14 October 2021). Figure 3 shows a fragment of the diagram of this ontology. For simplicity reasons and better appearance, the visualization is shortened. The four levels of abstraction are shown in the diagram: upper level includes the most abstract concepts that are taken from BFO. These concepts are colored green. The middle level consists of developed IIP concepts according to Tables 1 and 2. Nodes of this level are blue. Domain-specific concepts are yellow. These concepts were included in the ontology in the process of knowledge engineering with the expert. Finally, the bottom level is the data level where several individuals represented. With colored arrows the relations or object properties are shown. It is important to point out that these relationships are not domainspecific, but are defined in the IIP CRM. Therefore, it can be argued that the created hierarchy of concepts has been enriched by additional relations during the creation of individuals. This provides several benefits at once and shows the process of shaping knowledge graphs based on the developed CRM:

• Domain-specific concepts may be linked with CRM defined properties that make it easier to maintain the consistent semantics the knowledge graph, writing queries etc.

• Using the CRM defined concepts and properties it is possible to write semantic queries for domain-specific questions that provide interoperability and scaling the process models. An example of such a query will be given below.



It is necessary to comment on the use of reification for representation of a sequence of individual process steps. As explained earlier, despite a very simple container-like modeling patten, reification causes several problems including weak semantics of relations, necessity to write more complex and puzzled queries and some difficulties with calculations

of embeddings. Thus, the use of reification was not used. Table 3 contains an aggregated structure of the dataset for 5 years, which is used as an example in this article. The OWL file in turtle forma is available in the repository. Figure 4 shows a diagram illustrating how

Test 1 Test 3 Oper 1 Test 2 Oper 2 Oper 3 Test 4 Year Low Nitrogen 2015 Normal Normal High Nitrogen agrochemical agrophysical agrophysical 2016 High Nitrogen agrophysical Low Nitrogen agrochemical High Nitrogen agrophysical Low Nitrogen Low Nitrogen 2017 Normal agrochemical Low Nitrogen agrophysical Low Nitrogen agrochemical 2018 Normal Low Nitrogen Normal agrophysical High Nitrogen agrochemical agrochemical 2019 High Nitrogen agrochemical Normal agrochemical Low Nitrogen agrophysical Normal



Figure 4. An example of annotated and linked domain-specific data.

The second step of evaluation was query writing. As an example of a competency question, take the following "What methods of artificial modification of soil properties have been successfully performed under different alternatives of agrotechnical process development in the case where low nitrogen data have been collected? Successful options

Table 3. Structure of the illustrative dataset from the example.

domain-specific data were annotated and linked.

are those where in the process of applying these measures, low nitrogen data were no longer reported." The corresponding SPARQL-query using only CRM terms would look like this:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX owl: <http://www.w3.org/2002/07/owl#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> PREFIX ex: <http://purl.org/iip-srm.owl#>

```
SELECT ?y ?pi ?s ?p ?o
```

WHERE {

```
?s rdf:type/rdfs:subClassOf* <http://purl.org/iip-
    srm.owl#Step_of_Process>;
    ex:affects ?p;
    ex:performed_in ?pi;
    ex:date ?date.
?pi ex:year ?y.
?o ex:is_observation_for ?pi;
    ex:value 'Low Nitrogen';
    ex:date ?date o.
    FILTER ( ?date > ?date_o)
    FILTER NOT EXISTS {
    ?o1 ex:is_observation_for ?pi;
         ex:value 'Low Nitrogen';
         ex:date ?date_o1.
    FILTER (?date_o1 > ?date)
}
```

```
} ORDER BY ASC(?y) ASC(?date)
```

The result of this query includes a year of the process implementation, the process identification ("weat_prodiction_X" in the example), performed process steps, specific result achieved during the process and the name of property of soil measurement when 'Low Nitrogen' was observed. Figure 5 shows the results for querying the experimental KGs. In the results, we can see only those steps of processes implementation that after the detection of low nitrogen allowed to maintain it at normal or high levels (see Table 3). There are data available for 2015, 2018, and 2019. The years 2016 and 2017 were not included in the results of the query, because the implemented options of modifying soil properties did not lead to an improvement in measures.

| У | pi | S | p | 0 | |
|--------|-----------------------|---------------------------------------|---------------------|-------------|--|
| "2015" | wheat_production_2015 | distribution_over_soil_surface_2015_1 | content_of_nitrogen | test_2015_1 | |
| "2015" | wheat_production_2015 | ameliorative_measures_2015_2 | soil_texture | test_2015_1 | |
| "2015" | wheat_production_2015 | ameliorative_measures_2015_3 | soil_texture | test_2015_1 | |
| "2018" | wheat_production_2018 | distribution_over_soil_surface_2018_3 | content_of_nitrogen | test_2018_3 | |
| "2019" | wheat_production_2019 | ameliorative_measures_2019_3 | soil_texture | test_2019_3 | |
| | | | | | |
| | | Evecute | | | |
| | | Execute | | | |

Figure 5. The query results.

This example shows that the CRM provides the ability to answer specific questions in terms the expert knows. Meanwhile, the queries themselves can be written mainly using the CRM vocabulary, which makes it much easier to develop applications based on the KGs.

The example shows only a few data properties. In real practice, of course, there will be significantly more such properties, and they are important for the development of a valuable knowledge graph. However, since they are not part of the developed model, their number in the example does not have much importance. Hereby the given pattern provides a useful indication of how developed IIP CRM may be applied. The domain-specific data transformed using IIP CRM into a knowledge graph become harmonized, relevant, and significant information recourse. In addition, the mapping domain concepts to the CRM allows the achievement of both data structuring and enrichment that is essential for an individual process understanding.

6. Discussion

This paper has argued that the starting point for the creation of an efficient individualization of information processes should be development a relevant conceptual reference model. Such a model allows not only facilitate an access to heterogeneous data but also represent corresponding domain knowledge used for decision-making about process variants. Reference models are closely related to concept (ontological) and process modeling. Both are core elements for individualization task. Thus, in this work the abstract semantic reference model is represented as an ontology.

Designing ontologies is now a fairly well-developed methodology. There are enough different upper ontologies to represent completely abstract entities as well as personalities, organizations, manufacturing concepts, and many others. At the same time, upper-level ontologies remain a point of great interest to the community and their standardization proceeds, including at the institutional organization such as ISO. Examples are the standardization works of well-known and widely used ontologies such as Dublin Core (https://www.iso.org/standard/71341.html, accessed on 14 October 2021) or BFO (https://www.iso.org/standard/74572.html, accessed on 14 October 2021).

In this paper, BFO was chosen as the source of numerous concepts and patterns in the creation of the reference model. This choice is explained by the two levels of modeling in BFO which allow the separation of the reasoning at the level of concepts that have spatio-temporal boundaries and not tied to time or, in other words, processes and objects. The BFO ontology also defines the relationships between these two levels. The reference model also includes several concepts specific to IoT-enabled manufacturing processes, for which a mapping to ontologies such as MASON and GFO is made.

As a real-world example, the area of precise agriculture was considered. For this area, the use of both IoT technologies and new hybrid AI methods are promising. Although the implementation of IoT technologies in the processes of precise agriculture is already demonstrating some success, the use of AI methods meets barriers related to the complexity and diversity of data, as well as the lack of knowledge bases for the subject area. The proposed conceptual reference model aims to address both challenges by developing a knowledge graph that has this model as a conceptual schema.

Today, exploring ways to combine latent and model-based semantics is a topic of challenging research. In [34], opportunities and applicability of knowledge graphs to hybrid AI methods are discussed. All these approaches require the use of some graph of data as input data. In particular, Graph-To-Sequence models make it possible to determine sets of concepts relevant to specific subgraphs. In the task of individualization of processes, such methods may solve the problem of dynamic variability in process planning. This is an area requiring further research. Thus, our future work will focus on studying the features of knowledge graphs for individualized processes that affect the performance of hybrid AI methods, as well as adapting these methods to domain-specific requirements.

Funding: This research was partially funded by the Ministry of Education and Science of Russia No. 075-15-2020-805 2 October 2020.

Data Availability Statement: http://purl.org//iip-srm/ATO-example (accessed on 14 October 2021).

Acknowledgments: The study was partially financially supported by the Russian Federation (Agreement with the Ministry of Education and Science of Russia No. 075-15-2020-805, 2 October 2020).

Conflicts of Interest: The author declares no conflict of interest.

References

- 1. Florian, G.; van der Aalst, W.M.P.; Jansen-Vullers, M.H. Configurable process models—a foundational approach. In *Reference Modeling*; Physica-Verlag HD: Heidelberg, Switzerland, 2007; pp. 59–77.
- 2. Suri, K.; Gaaloul, W.; Cuccuru, A. Configurable IoT-aware allocation in business processes. In *International Conference on Services Computing*; Springer: Cham, Switzerland, 2018; pp. 119–136.
- 3. OASIS SOA Reference Model. OASIS SOA Technical Committee; OASIS Open: Burlington, MA, USA, 2006.
- 4. Davies, I.; Green, P.; Rosemann, M.; Indulska, M.; Gallo, S. How do practitioners use conceptual modeling in practice? *Data Knowl. Eng.* **2006**, *58*, 358–380. [CrossRef]
- de Vries, M.; van der Merwe, A.; Kotze, P.; Gerber, A. A Method for Identifying Process Reuse Opportunities to Enhance the Operating Model. In Proceedings of the 2011 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 6–9 December 2011.
- 6. Becker, J.; Meise, V. Strategy and Organizational Frame. In *Process Management. A Guide for the Design of Business Processes*; Becker, J., Kugeler, M., Rosemann, M., Eds.; Springer: Berlin, Germany, 2011; pp. 91–132.
- Becker, J.; Schütte, R. Reference Information Systems for Retail: Definition, Use and Recommendations for Design and Company-Specific Adaption of Reference Models; Wirtschaftsinformatik; Springer: Berlin/Heidelberg, Germany, 1997; pp. 427–448.
- 8. Rehse, J.R.; Fettke, P.; Loos, P. An Execution-Semantic Approach to Inductive Reference Model Development; Association for Information System: Atlanta, GA, USA, 2016.
- 9. Ahlemann, F. Towards a conceptual reference model for project management information systems. *Int. J. Proj. Manag.* 2009, 27, 19–30. [CrossRef]
- 10. Froese, T. Integrated Computer-Aided Project Management through Standard Object-Oriented Models; Center for Integrated Facility Engineering: Stanford, CA, USA, 1992.
- 11. Schlagheck, B. Objektorientierte Referenzmodelle für das Prozess- und Projektcontrolling: Grundlagen—Konstruktion—Anwendungsmöglichkeiten; Gabler: Wiesbaden, Germany, 2000.
- 12. De Nicola, A.; Lezoche, M.; Missikoff, M. An Ontological Approach to Business Process Modeling. In Proceedings of the 3rd Indian International Conference on Artificial Intelligence, Puna, India, 17–19 December 2007; pp. 1794–1813.
- 13. Merunka, V. Object-oriented process modeling and simulation-BORM experience. Trakia J. Sci. 2010, 8, 71–87.
- Rassler, J.; Anderl, R. PML, an Object Oriented Process Modeling Language. In Computer-Aided Innovation (CAI): IFIP 20th World Computer Congress, Proceedings of the Second Topical Session on Computer-Aided Innovation, WG 5.4/TC, Milano, Italy, 7–10 September 2008; Springer: Berlin, Germany, 2008.
- 15. Mouromtsev, D. Models and methods of e-learning individualization in the context of ontological approach. *Ontol. Des.* **2020**, *10*, 34–49. [CrossRef]
- 16. Frey, J.; Müller, K.; Hellmann, S.; Rahm, E.; Vidal, M.E. Evaluation of metadata representations in RDF stores. *Semant. Web* 2019, 10, 205–229. [CrossRef]
- Hernández, D.; Hogan, A.; Riveros, C.; Rojas, C.; Zerega, E. Querying wikidata: Comparing sparql, relational and graph databases. In Proceedings of the International Semantic Web Conference, Kobe, Japan, 17–21 October 2016; Springer: Cham, Switzerland, 2016; pp. 88–103.
- Sassanelli, C.; Seregni, M.; Hankammer, S.; Cerri, D.; Terzi, S. The role of Internet of Things (IoT) technologies for individualisation and service quality of a PSS. In Proceedings of the 21st Summer School Francesco Turco 2016, Naples, Italy, 13–15 September 2016; AIDI-Italian Association of Industrial Operations Professors: Naples, Italy, 2016.
- 19. Available online: https://blogs.oracle.com/marketingcloud/personalization-vs-individualization:-what-they-are-and-how-to-use-them (accessed on 14 October 2021).
- 20. Available online: https://www.wipro.com/blogs/sarath-kumar-ganesan/the-shift-from-personalization-to-hyper-individualization (accessed on 14 October 2021).
- 21. Huang, J.; Zhang, J.; Chang, Q.; Gao, R.X. Integrated process-system modelling and control through graph neural network and reinforcement learning. *CIRP Ann.* **2021**, *70*, 377–380. [CrossRef]
- Banarescu, L.; Bonial, C.; Cai, S.; Georgescu, M.; Griffitt, K.; Hermjakob, U.; Knight, K.; Koehn, P.; Palmer, M.; Schneider, N. Abstract meaning representation for sembanking. In Proceedings of the 7th linguistic annotation workshop and interoperability with discourse, Sofia, Bulgaria, 8–9 August 2013; pp. 178–186.
- 23. Dwivedi, V.P.; Bresson, X. A Generalization of Transformer Networks to Graphs. arXiv 2020, arXiv:2012.09699.
- 24. Heller, B.; Herre, H. Ontological categories in GOL. Axiomathes 2004, 14, 57–76. [CrossRef]
- 25. Arp, R.; Smith, B.; Spear, A.D. Building Ontologies with Basic Formal Ontology; MIT Press: Cambridge, MA, USA, 2015.
- 26. Zemach, E. Four Ontologies. J. Philos. 1970, 23, 231-247. [CrossRef]
- 27. The BFO 2 Reference. Available online: http://purl.obolibrary.org/obo/bfo/2012-07-20/Reference (accessed on 14 October 2021).
- 28. Herre, H. General Formal Ontology (GFO): A foundational ontology for conceptual modelling. In *Theory and Applications of Ontology: Computer Applications*; Springer: Dordrecht, The Netherlands, 2010; pp. 297–345.
- 29. Lemaignan, S.; Siadat, A.; Dantan, J.Y.; Semenenko, A. MASON: A proposal for an ontology of manufacturing domain. *IFAC Workshop Distrib. Int. Syst.* **2006**, 2006, 195–200.
- 30. DCMI Metadata Terms. 2020. Available online: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/ (accessed on 14 October 2021).

- Niles, I.; Pease, A. Towards a Standard Upper Ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, ME, USA, 17–19 October 2001; Welty, C., Smith, B., Eds.;
- 32. Grenon, P.; Smith, B. SNAP and SPAN: Towards Dynamic Spatial Ontology. Spat. Cogn. Comput. 2004, 4, 69–104. [CrossRef]
- Halpin, H.; Hayes, P.J.; McCusker, J.P.; McGuinness, D.L.; Thompson, H.S. When owl:sameas isn't the same: An analysis of identity in linked data. In *The Semantic Web—ISWC 2010*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; pp. 305–320.
- Bonatti, P.A.; Decker, S.; Polleres, A.; Presutti, V. Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web (Dagstuhl Seminar 18371); Dagstuhl Reports; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl: Wadern, Germany, 2019; Volume 8, p. 9.