SIMISS: A Model-Based Searching Strategy for Inventory Management Systems

Yan Tang Demey and Mikael Wolff

***Abstract*—Inventory management is critical in human space flight operations. Currently, we use the inventory management system (IMS) in keeping track of items on the International Space Station (ISS). One challenge is to discover lost or wrongly placed items when IMS fails to discover them due to human factors. In this paper, we will illustrate a model-based searching strategy called semantic inventory management for ISS (SIMISS), with which possible locations of lost items will be calculated based on contextual features in three dimensions: 1) *spatial*; 2) *temporal*; and 3) *human*. It contains ontologies, databases, machine learning algorithms, and ubiquitous client applications. We have implemented and tested SIMISS with the sample data from IMS, operation data files and onboard short term plan experiments have been carried out in a set of simulation scenarios.**

***Index Terms*—Big data analysis, fact-based modeling (FBM), Internet of Things (IoT), radio frequency identification (RFID).**

I. Introduction

INVENTORY management is essential in many domains, such as production and supply chain management. In the space business, inventory management is critical and directly affects the effectiveness and efficiency of any human spaceflight missions. As indicated in the recommended practice red book drafted by the consultative committee for space data systems (www.ccsds.org) [1], improperly substituted items and early depletion of items can be catastrophic.

Inventory management system (IMS) is used to trace items on the International Space Station (ISS). Names of inventory items are consistent with the dictionary of operations nomenclature (OpNom) from National Aeronautics and Space Administration Brown *et al.* [2] stated that the current IMS has been successful in keeping track of about 97% of over 10 000 items. Approximately 3% of items are considered lost due to human factors. For instance, a crew member may forget to stow a tool after a procedure, move an item without notifying the ground due to the lack of time, or be unaware of the situation that the IMS database is not updated successfully.

Manuscript received April 22, 2015; revised November 15, 2016; accepted November 28, 2016. Date of publication December 9, 2016; date of current version February 8, 2017. This work was supported by the ESA under the Research Fellowship Program (RFP). *(Corresponding authors: Yan Tang Demey; Mikael Wolff.)*

Y. T. Demey is with the European Patent Office, Department of Requirements Engineering and Solution Design (Search and Knowledge Management IM2.8.3.3), 2288 EE Rijswijk, The Netherlands (e-mail: ytang@epo.org).

M. Wolff is with the European Space Agency, ESA/ESTEC/TEC-SWM, 2201 AZ Noordwijk, The Netherlands (e-mail: mikael.wolff@esa.int).

Digital Object Identifier 10.1109/JIOT.2016.2638023

IMS uses near-distance technologies (i.e., tags with barcodes and barcodes scanners) for operations on ISS. This subsystem is called barcode inventory tracking system (BITS). Fink *et al.* [3] proposed to use wireless technologies [e.g., radio frequency identification (RFID)], together with middleware, to increase situational awareness and autonomy of IMS. Till today, BITS is still the only available item tracking system for all the European Space Agency (ESA) Columbus onboard missions. In this paper, we will present a searching strategy that has been studied using IMS and BITS data. This strategy is generic so that it is ready to be tested with the future technologies (e.g., as proposed in [3]) as well.

We take an approach of semantic technologies (more precisely, ontologies [4], [5] and a machine learning algorithm. The strategy is called semantic inventory management for ISS (SIMISS). It constructs a virtual inventory that is completely visible to onboard crew members.

This paper is organized as follows. Section II contains the background and related work of this paper. In Section III, we will illustrate how SIMISS is designed. We will demonstrate the empirical study in Section IV. The conclusions are drawn in Section V.

II. Background and Related Work

There are few literature and research that cover automatic (lost) item discovery in inventory management. The main research stream focuses on the development of inventory management policies and supply chain support, aiming to allow faster response to market demand changes whilst maintaining lower inventory levels [6]. The challenge, which IMS is dealing with, is uncommon and probably unique, seeing that: 1) many short duration missions to ISS involve very limited restocking or resupply logistics and 2) the resupply logistics of most long duration missions are expensive and infrequent. Instead, the challenge is to *discover lost items*. The reasons are justified as follows.

1) An item that is not traceable by IMS (i.e., a "lost" item) still *physically* resides in ISS because all the ISS modules are closed. To keep a module neat and organized, especially when it is a science laboratory, is essential. Therefore, lost items need to be discovered and restored back to right racks.
2) As mentioned, most short duration missions do not involve restocking or resupply logistics. When a lost item happens to be mission critical and unreplaceable, it will lead to a mission failure. The consequence will be

propagated when there are many missions that depend on the outcome of this mission. Hence, to be able to discover lost important items is often necessary.

Before knowing how to discover a lost item, we first need to know how information and data of items is structured. A traditional way[1] is to categorize items and search (or manually browse) based on keywords. Categorization is a way to simplify our interactions with systems. Rosch [7] has illustrated two basic principles of categorization: 1) principle of *maximum information* and 2) principle of *least cognitive effort*. All the members (i.e., items in an inventory) in the same category shall have the same level of abstraction.

Similar approaches are classification-based, categorization-based, or taxonomy-based IMSs. Categorization and classification are obviously similar but are essentially different [8]. The process of classification is rigorous and mandates an entity to be either is or is not a member of a class. The process of categorization is flexible and draws conciliatory associations between entities. Taxonomy is a science including studies of principles that underline classifications. Classification-based and categorization-based approaches can be found in [9]–[12]. And taxonomy-based methods are covered in [13] and [14].

Recently, ontology-based IMSs draw research attention. Compared to classifications and taxonomies, ontologies contain richer semantics for modeling entities (i.e., concepts) and relations between entities (i.e., formal conceptual relations) in a domain. Ontologies enhance system interoperability and enable reuse of domain knowledge. Ontology-based approaches to inventory management can be found in [15] and [16].

Our approach is ontology-based and the ontology is migrated from categories and taxonomies. Unlike the related work, we follow a methodological principle of fact-based modeling (FBM) approach [17] in order to properly capture domain semantics, which can be published, stored and exchanged in popular ontology languages, such as Web ontology language (OWL/OWL2) and resource description framework.

FBM is a family of information modeling dialects, which include natural language information analysis method (NIAM) [18], cognition enhanced NIAM [19], object rule modeling language [20], fully communication oriented information modeling [21], and developing ontology-grounded methods and applications [22], [23]. The methodological principle of FBM is to extract information from plausible facts in a given domain by adhering to the conceptualization and 100% Principles of ISO TR9007 [24].

In the next section, we will illustrate the FBM model-based SIMISS strategy.

III. SIMISS

SIMISS contains a knowledge base, a database, a machine learning algorithm, and ubiquitous client applications. The knowledge base contains ontologies.

Ontologies in SIMISS can be versatile, which requires a proper architectural design. In [25] and [26], ontologies are layered and modularized based on the coverage of concepts and abstraction levels. Such layers help application development with scalability, extensibility, and flexibility.

The ontology base of SIMISS contains a global ontology and several domain ontologies. Global ontology is similar to “foundational ontologies” in [26], “base ontologies” [25], “core reference ontology” in [27], or “generic ontologies” or “top-level ontologies” as mentioned in [26]. A global ontology is generic to all the known fields.

A domain ontology contains formalization of a domain of interest. It is specific to a particular field. A domain ontology can contain another domain ontology (sometimes called “subdomain ontology” or a model in a domain ontology that is restricted to a certain context).

The concept of context is important in context-aware systems. In [28], a context is defined as circumstances that forms an event in the concepts of person, community, time, object, activity, and location.

We simplify our settings into the following implementation assumptions for a practical reason.

1) There is *only one ontology* within one iteration of system design.
2) This ontology contains *context*-specific models based on the work in [28].

In Section III-A, we will illustrate the ontology (in particular, the inventory model, behavior model, and procedure model in the ontology).

SIMISS uses the population of the ontology to store training data for machine learning algorithms. Note that SIMISS is not limited to certain types of machine learning algorithms. However, we select the Iterative Dichotomiser 3 (ID3) [29] algorithm—a classic decision tree classifier—in this paper for the purpose of demonstrating how SIMISS works. We will discuss it in Section III-B.

A. Ontologies in SIMISS

To properly study *context*s [30] is important in context-aware systems during different evolutionary periods, namely the epochs of *distributed computing, [31], ubiquitous systems, [32], pervasive computing, [33], and Internet of Things (IoT) [34]*. In this section, we will illustrate models that contain three dimensions: 1) spatial; 2) temporal; and 3) human dimensions, which are the basic constituents for context analysis. It is based on the context ontology in [28], where the key concepts are illustrated as follows.

1) *Person* is an individual of human beings.
2) *Community* is any group of human being collectively. Each group is specified with a common role that all the members play within a context.
3) *Time* is an identifiable instance or a period of a moment of an action executed by a person.
4) An *object* is a tangible thing used by a person.
5) An *activity* is executed by a person through space and time.
6) A *location* is a point or extent in space.

The three dimensions are interlinked. We group the concepts of “person” and “community” in the human dimension,

[1]Note that this is also the way how item information is stored in IMS.

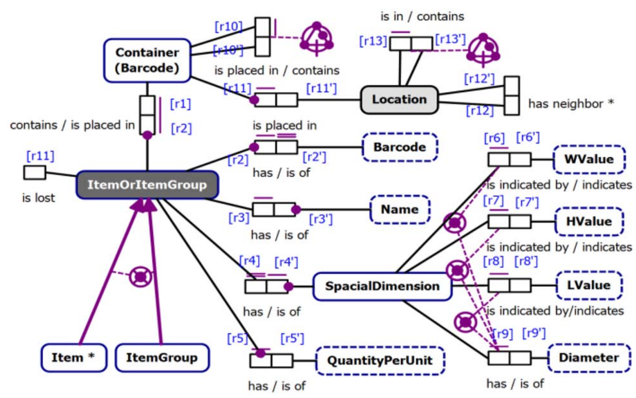


Fig. 1. Inventory model (partial view) in NORMA.

which will be modeled as the concept "CrewOrCrewMember" in our ontology. In the temporal dimension, "time" is further specified in the ontology using "duration" and "timestamp." In the spatial dimension, we reuse the concept "location." The "ItemOrItemGroup" concept is a subtype of "object." It links to a spatial dimension (i.e., its physical mass and the location of its container). We further specify a component of "activity"—procedure—in temporal and spatial dimensions (i.e., when and where an item or item group was used last time in a procedure). We will elaborate on the modeling issue later in this section.

We use the FBM approach to modeling ontologies in this paper. According to the standardization working draft [17], in the FBM terminology, concepts (i.e., *object types*) are either entity types or value types. An *entity type* is defined as an object type, each of whose instances is an *entity* (i.e., an object that is referenced by a definite description that relates it to other objects). A *value type* is an object type, each of whose instances is a (domain) value. A relation between concepts is called *predicate*. A *role* is a part played by an object in a fact. A combination of concepts and relations are formalized in FBM as *fact type*s. An ontological axiom is presented as an FBM *constraint*, which is a restriction on what states or transitions of the *fact base* (i.e., a set of plausible facts) are possible or permitted.

An FBM model is represented in a graph and texts in a controlled natural language (also called "verbalization" in [35]–[37]).

Recently, FBM verbalization is applied in requirement engineering for modeling requirements of space systems [38]. Following this idea, we have modeled an inventory model that covers the data semantics in IMS by taking the relevant requirements into consideration. Fig. 1 shows a partial view of this ontology using FBM graphical notations.

An entity type (with a reference mode) is depicted by a named soft rectangle, e.g., in Fig. 1, "container" is an entity type and "(barcode)" means that each container is referred by a barcode. A value type is rendered as a named, dashed, soft rectangle, e.g., "name." A predicate is represented as an ordered set of one or more boxes with a predicate reading, e.g., the predicate indicated with "has" between the entity type "SpatialDimension" and the value type "diameter." A subtyping is modeled as an arrow-tipped bar, e.g., "item" is a subtype of ItemOrItemGroup. A simple mandatory constraint is indicated by a solid dot between a role and an object type, e.g., each ItemOrItemGroup is placed in at least one container. An external mandatory constraint is a dot that connects at least two roles. A circled cross indicates an exclusion constraint. An external mandatory can be combined with an exclusion constraint, the result of which is an *exclusive-or* constraint, e.g., the exclusive-or constraint between item and "ItemGroup" being the two subtypes of ItemOrItemGroup. A uniqueness is indicated as a bar over a role, e.g., ItemOrItemGroup has at most one barcode. When a bar is a double lined, it is a preferred identifier indicator, e.g., barcode is the preferred identifier of ItemOrItemGroup. The circled triangle with dots in Fig. 1 indicates a ring constraint. In particular, this is a constraint of strongly intransitive and acyclic. The square brackets denote role names. For instance, r_1 is the name of the role "contains" played by container toward ItemOrItemGroup; r'_1 is the name of the role "is placed in" played by ItemOrItemGroup toward container.

For a purpose of presenting shared concepts and predicates between models, we decorate shared concepts in a color, e.g., ItemOrItemGroup in Fig. 1.

With regard to the FBM verbalization, for instance, for the exclusive-or constraint between item and ItemGroup, it is verbalized as follows.

For each item or item group, exactly one of the following holds:
that item or item group is an item;
that item or item group is an item group.

We refer to [38] for the details of verbalization, with which our domain experts from IMS could validate the ontology.

FBM models can be completely mapped to description logic (DL) [39] axioms. We choose the $\{\mathcal{S}\}\mathcal{OIQ}(D)$ DL dialect in this paper. We represent some representative patterns in Table I. Patterns that include previous patterns, the axioms of the previous ones are repeated. For instance, pattern 2 is not repeated in patterns 4 and 6. Note that role r''_{13} is an implicit role defined as a property chain of r_{13} of two steps. It is used to restrict the population of a transitivity relation. Note also that data types (e.g., in patterns 3 and 11) are not considered as "purely" logical axioms [40].

In FBM, we use * to indicate validation rules (called "derivation rules" in FBM) that are not graphically represented. For example, the validation rule for item (indicated with * in Fig. 1) is formalized as the axiom of pattern 17 in Table I. It is verbalized as follows.

Each item is an item or item group, the quantity per unit of which shall be 1.

The validation rule for "location has neighbor location" in Fig. 1 cannot be formalized as a DL axiom. We use FOL to formalize it as follows:

$$\{\forall l_1 l_2 l_3 | [r_{13}, r'_{13}](l_1, l_3) \cap [r_{13}, r'_{13}](l_2, l_3) \rightarrow [r_{12}, r'_{12}](l_1, l_2)\}$$

where l_1, l_2, l_3 are three domain objects of location (i.e., three instances in the population of location), $[r_{13}, r'_{13}]$

TABLE I
FORMALIZATION IN DL

ID	Pattern	DL Axiom Example
1	*Role*	$\exists r_1.\top \sqsubseteq Container$
2	*Role pair of binary fact types*	$r_1^{-} \equiv r$
3	*Role of unary fact types*	$ItemOrItemGroup \sqsubseteq \forall r_{11}.\boldsymbol{Boolean}$
4	*Predicate*	$Container \sqsubseteq \forall r_1.ItemOrItemGroup$
5	*Subtyping*	$Item \sqsubseteq ItemOrItemGroup$
6	*Simple uniqueness*	$ItemOrItemGroup \sqsubseteq \leq 1 r_3.Name$
7	*Spanned uniqueness*	$Container \sqsubseteq \forall r_1.ItemOrItemGroup$ $ItemOrItemGroup \sqsubseteq \forall r_1'.Container$
8	*Simple mandatory*	$Name \sqsubseteq \exists r_3'.ItemOrItemGroup$
9	*Exclusive-or between subtypes*	$ItemOrItemGroup \equiv Item \sqcup ItemGroup$ $Item \sqcap ItemGroup \equiv \bot$
10	*Exclusive-or between fact types*	$SpacialDimension \equiv \forall r_6.WValue$ $\sqcup \forall r_9.Diameter$
11	*Known value type*	$ItemOrItemGroup \sqsubseteq \boldsymbol{Integer}$
12	*Extended value type*	$TimeStampe \sqsubseteq \boldsymbol{Timestampe(Minutes)}$
13	*Ring (Intransitive)*	$r_{13}.r_{13} \equiv \boldsymbol{r_{13}}''$ $Location \sqsubseteq \neg(\forall r_{13}''.Self)$
14	*Ring (Acyclic)*	$Location \sqsubseteq \neg(\forall r_{13}.Self)$
15	*Subtype derivation with value constraint*	$Item \sqsubseteq ItemOrItemGroup \sqcap \forall r_5.\{1\}$
16	*External uniqueness*	$ExpOfProc \equiv \boldsymbol{Key}(b_1, b_2', b_3', b_4')$
17	*Subtype derivation with negation*	$MajorStep \sqsubseteq Step \sqcap \neg\forall r_{12}'.Step$

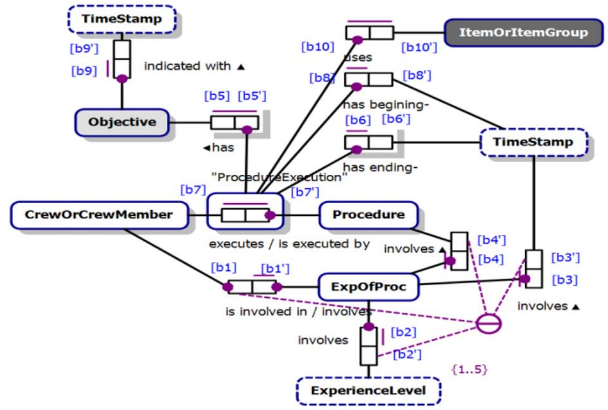


Fig. 2. Behavior model of crew or crew member (partial view).

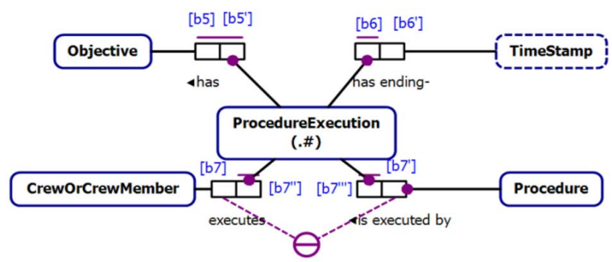


Fig. 3. Transformed objectification.

and $[r_{12}, r'_{12}]$ are the role pairs that can be found in Fig. 1, $[r_{13}, r'_{13}](l_1, l_3)$ indicates that l_1, l_3 play the role pair $[r_{13}, r'_{13}]$ (i.e., an instance in the population of "location is in/contains location"). N

This axiom is verbalized as follows.

If location$_1$ is in location$_3$, and location$_2$ is in location$_3$, then location$_1$ shall have neighbor location$_2$.

The second model used in this strategy is called behavior model of crew and crew member. Human behavior and its impact on the provision of onboard missions are considered, yet infrequently studied. In order to build the behavior model of crews or crew members, we studied two aspects of human behaviors: 1) *human reliability* and 2) *human naturalistic decision making* (NDM). The former studies unexpected behaviors from a crew or crew member, e.g., moving an item without notifying the ground or updating the onboard IMS database. The latter covers the following case: a crew or crew member makes decisions or finds alternative solutions when an onboard activity cannot not be executed fully according to the procedure.

A number of studies on *human reliability* have been conducted in the context of work tasks of process management based on historical data [41], [42]. Accordingly, we consider historical data of a crew or crew member with regard to procedure execution as one of the important factors in the behavior model, which is formalized as the concept "ProcedureExecution" and its relationships with other concepts in Fig. 2. Concepts with a gray background are the ones that have been presented earlier in this paper.

The study of NDM has been evolved to describe how human beings make decisions in the real world [43], [44]. Among the common environmental factors of NDM discussed in the literature, we have selected *time stress*, *changing objectives*, and *experiences* as the most applicable ones and modeled in Fig. 2. Other factors like ill-structured or incomplete/faulty documents, uncertainty, risk and interactions between individuals are important in the human space flight, but of less importance in our discussion, and thus neglected.

The pattern and example of an external uniqueness, which is graphically represented as a circled bar ⊖, are recorded as pattern 15 in Table I. External uniqueness constraints are comparable to OWL keys. As data types, keys are not considered as purely logical axioms [40].

A named, soft rectangle enclosing the predicate shape indicates an *objectification*, which turns a predicate into an object type. It can be directly formalized in a second-order logic. However, since DL is first-order logic, we use the method of model transformation to replace the objectification with other constrained fact types. Fig. 3 shows how the objectification ProcedureExecution and the predicates indicated by $b5$, $b5'$, $b6$, $b6'$, $b7$ and $b7'$ are transformed. The roles $b7''$ and $b7'''$ restricted by an exactly-one constraint are added. The spanned internal uniqueness constraint on $(b7.b7')$ in Fig. 2 is transformed into an external uniqueness constraint on the same role pair.

The last model is onboard activity procedure. Procedures are documented as manuals for human spaceflight missions at ESA. Such a document is called operation data file (ODF). We take an approach of brainstorming and text analysis methods as suggested in [23] and [38] for the model creation. A partial view of this model is illustrated in Fig. 4.

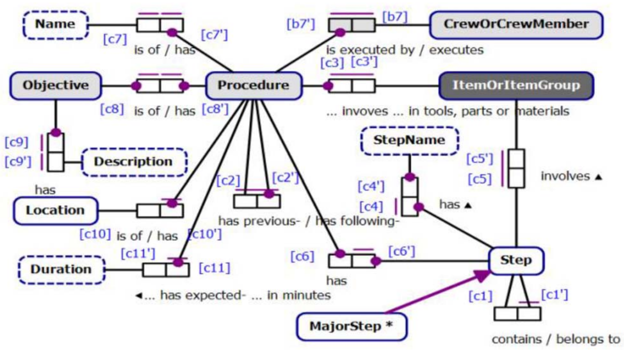


Fig. 4. Activity procedure model (partial view).

TABLE II
KEY ROLES THAT CONTRIBUTE THE THREE DIMENSIONS

Dimension / Model	Spatial	Temporal	Human
Fig. 1	$[r_1, r_1']$, $[r_4, r_4']$, $[r_6, r_6']$, $[r_7, r_7']$, $[r_8, r_8']$, $[r_9, r_9']$, $[r_{10}, r_{10}']$	N/A	N/A
Fig. 2	N/A	$[b_3, b3']$, $[b_6, b_6']$, $[b_8, b_8']$, $[b9, b9']$	$[b_1, b_1']$, $[b_2, b_2']$, $[b_7, b_7']$
Fig. 3	$[c_{10}, c_{10}']$	$[c_2, c_2']$	$[b7, b7']$

A mapping pattern from FBM to DL that is specific in this model is the validation rule of "MajorStep." It is recorded as the example of pattern 16 in Table I. And the verbalization is illustrated as follows.

Each major step is a step, where that step shall not belong to another step.

A procedure may have previous and following procedures (see the predicate indicated by roles $[c_2, c_2']$ in Fig. 4). The population is currently stored in ESA's onboard procedure library.

Table II shows key roles of the models in Figs. 1, 2, and 4 in the dis cussed three dimensions. The population of these key roles are included in the training data set for the algorithm that will be discussed in the next section.

B. Applied Iterative Dichotomiser 3

SIMISS uses machine learning algorithms for mining or clustering data that refer to searching tasks.

A class machine learning algorithm is called ID3, which was introduced by Ross [29]. It uses *information gain* (i.e., Kullback–Leibler divergence [45]) for constructing a decision tree by employing a top-down, greedy search through the space of possible branches without backtracking.

An extension to ID3 is C4.5 [46]. Unlike ID3, which allows only distinct values for attributes, C4.5 also allows continuous values. Another known extension is called classification and regression trees [47], which only builds decision trees that are binary.

In principle, SIMISS is not restricted to a particular algorithm. In this paper, we use the "ID3"—a classic decision tree-based algorithm—for demonstrating SIMISS.

ID3 is calculated based on the *information entropy* from a prior state.

Given a training data set S, its information entropy, $H(S)$, is calculated as shown in (1), where p_i is the probability of decision i

$$H(S) = \sum_i -p_i \log_2 p_i. \tag{1}$$

$H(S)$ is used to measure the impurity level of the training data set S. The higher the value of $H(S)$, the more the information content.

Let a denotes an attribute. The information gain for a over S, denoted as $IG(S, a)$, is calculated as follows:

$$IG(S, a) = H(S) - H(S|a). \tag{2}$$

$IG(S, a)$ is used to calculate expected reduction caused by partitioning the training data set S according to the attribute a. The attribute with the highest information gain is designated as a root node.

We denote P_{last} as the procedure where the lost item or item group was used for the very last time and $P_{\text{last}} \cdot cr$ as the crew or crew member who executed P_{last}. The assumed locations of the lost items follows in the following five decisions.[2]

1) *Loc1:* location of P_{last}.
2) **Loc2:* location of the procedures, which have been executed C_1 minutes after P_{last} by $P_{\text{last}} \cdot cr$ and which require similar tools.
3) *Loc3:* neighbor location of P_{last}.
4) **Loc4:* one of the neighbor locations of the procedures, which have been executed within C_1 minutes after P_{last} by $P_{\text{last}} \cdot cr$ and which require similar tools.
5) *Loc5:* random location (i.e., the location that is not in Loc1, 2, 3, or 4).

A neighbor location is a location of a container that is in the same place of the container of the discussed item. For example, the item called "21,5" Handrail" (identified by the bar code "00140444J") is stowed in the container "COL1D1_Rack Front," which is physically resides at the place "D Racks Front." D Racks Front contains two other containers ("COL1D2_Rack Front" and "COL1D3_Rack Front"). The locations of these two containers are considered as the neighbor locations.

The five decisions cover two types of assumptions. Loc1 and Loc3 cover the assumption that a crew or crew member forgot the lost item or item group at the place, where the last procedure that uses this item or item group was executed. Loc2 and Loc4 cover the assumption that a crew or crew member did not stow the item or item group after executing the last procedure that uses this item or item group; instead, he kept it for the follow-up procedures that require similar tools.

To simplify our use case, we will use the identifiers of containers to identify the container locations in the rest of this paper.

We use three classification attributes as follows.

1) a_1: Item size.
2) a_2: Experience level of $P_{\text{last}} \cdot cr$ with regard to P_{last}. Its value ranges from 1 to 5 (see Fig. 2).
3) a_3: Actual versus expected durations of P_{last}.

[2]A decision annotated with * means that it is dependent on some parameters. For instance, Loc4 depends on C_1. Note that we use C_i to denote parameters in this paper.

Attribute values can be fuzzy. For example, the values of a_1 can be "small," "medium," or "large." a_1 is small if it meets the following condition[3]:

$$\{\exists x_1x_2x_3x_4 | [r_6, r'_6](x_4, x_1) \cap [r_7, r'_7](x_4, x_2) \times \cap [r_8, r'_8](x_4, x_3) \cap x_1 \le C_2 \cap x_2 \le C_3 \cap x_3 \le C_4\} \ne \emptyset.$$

Or

$$\{\exists x_1x_2 | [r_9, r'_9](x_2, x_1) \cap x_1 \le C_5\} \ne \emptyset$$

a_1 is medium if

$$\{\exists x_1x_2x_3x_4 | [r_6, r'_6](x_4, x_1) \cap [r_7, r'_7](x_4, x_2) \cap [r_8, r'_8](x_4, x_3) \cap x_1 > C_2 \cap x_1 \le C_6 \cap x_2 > C_3 \cap x_2 \le C_7 \cap x_3 > C_4 \cap x_3 \le C_8\} \ne \emptyset.$$

Or

$$\{\exists x_1x_2 | [r_9, r'_9](x_2, x_1) \cap x_1 > C_5 \cap x_1 \le C_9\} \ne \emptyset$$

a_1 is large if

$$\{\exists x_1x_2x_3x_4 | [r_6, r'_6](x_4, x_1) \cap [r_7, r'_7](x_4, x_2) \cap [r_8, r'_8](x_4, x_3) \cap x_1 > C_6 \cap x_2 > C_7 \cap x_3 > C_8\} \ne \emptyset.$$

Or

$$\{\exists x_1x_2 | [r_9, r'_9](x_2, x_1) \cap x_1 > C_9\} \ne \emptyset.$$

Suppose that we use fuzzy values like "about same," "much longer" and "much shorter" for a_3. The value of a_3 is about same if

$$\{\exists x_1x_2x_3x_4x_5y | [b_6, b'_6](y, x_1) \cap [b_8, b'_8](y, x_2) \cap x_3 = x_1 - x_2 \cap [b_7''', b'_7](y, x_5) \cap [c_{11}, c'_{11}](x_5, x_4) \cap (x_4 = x_3 \cup x_4 > x_3 \cap x_4 - x_3 \le C_{10} \cup x_4 < x_3 \cap x_3 - x_4 \le C_{10})\} \ne \emptyset.$$

The value of a_3 is much longer if

$$\{\exists x_1x_2x_3x_4x_5 | [b_6, b'_6](y, x_1) \cap [b_8, b'_8](y, x_2) \cap x_3 = x_1 - x_2 \cap [b_7''', b'_7](y, x_4) \cap [c_{11}, c'_{11}](x_4, x_5) \cap x_3 > x_5 \cap x_3 - x_5 > C_{10}\} \ne \emptyset.$$

The value of a_3 is much shorter if

$$L\{\exists x_1x_2x_3x_4x_5 | [b_6, b'_6](y, x_1) \cap [b_8, b'_8](y, x_2) \cap x_3 = x_1 - x_2 \cap [b_7''', b'_7](y, x_4) \cap [c_{11}, c'_{11}](x_4, x_5) x_3 < x_5 \cap x_5 - x_3 > C_{10}\} \ne \emptyset.$$

A small example of training data set is illustrated in Table III. To simplify our use case, the experience levels of our crew or crew member for all the procedures are in {3, 4, 5}.

Let us apply (1) for calculating the information entropy, as shown below

$$H(S) = \left(-\frac{2}{10} * \log_2 \frac{2}{10}\right) * 5 \approx 2.32193.$$

[3]All the roles *ri* (e.g., *r*6) can be found in Fig. 1, *bi* (e.g., *b*8) in Fig. 2 and *ci* (e.g., *c*11) in Fig. 4.

TABLE III
TRAINING DATA SET EXAMPLE: S-SMALL, M-MEDIUM, L-LARGE; AS-ABOUT SAME; ML-MUCH LONGER; AND MS-MUCH SHORTER

ID	a_1	a_2	a_3	Decision
1	S	3	ML	Loc1
2	S	4	MS	Loc2
3	M	3	MS	Loc3
4	M	3	ML	Loc4
5	L	4	AS	Loc5
6	L	5	MS	Loc1
7	M	5	AS	Loc2
8	M	5	ML	Loc3
9	S	4	ML	Loc4
10	L	3	AS	Loc5

When considering a_1 as the tree root. For the branch $a_1 = S$, we get

$$-\frac{1}{3} * \log_2\frac{1}{3} - \frac{1}{3} * \log_2\frac{1}{3} + 0 - \frac{1}{3} * \log_2\frac{1}{3} + 0 \approx 1.58496.$$

For the branch $a_1 = M$, we get

$$0 - \frac{1}{4} * \log_2 \frac{1}{4} - \frac{2}{4} * \log_2 \frac{2}{4} - \frac{1}{4} * \log_2 \frac{1}{4} + 0 = 1.5.$$

And for the branch $a_1 = L$, we get

$$-\frac{1}{3} * \log_2 \frac{1}{3} + 0 + 0 + 0 - \frac{2}{3} * \log_2 \frac{2}{3} \approx 0.9183.$$

Therefore,

$$H(S|a_1) \approx \frac{1.58496 * 3 + 1.5 * 4 + 0.9183 * 3}{10} \approx 1.35098.$$

Let us apply (2), we get

$$IG(S, a_1) \approx 2.32193 - 1.35098 = 0.97095.$$

By applying (1) and (2) for a_2 and a_3, we get

$$IG(S, a_2) \approx 0.57095$$
$$IG(S, a_3) \approx 0.97095.$$

Since $\max_{i\in\{1,2,3\}} IG(S, a_i) = 0.97095$, either $a_1$or a_3 can be chosen to become the tree root. Suppose a_1 is the root. We continuously apply (1) and (2) to form the subtrees. Fig. 5 shows the result. It is a decision tree constructed from the training data in Table III.

Each tree node is an attribute annotated by its value of information gain over the sample (sub-)set. The values of the attribute are represented as arcs.

When we need to find a lost item, we need to gather the item data that are stored in the population of our ontology and feed them to the decision tree.

Both of the MIS data set and the ODF data set complies with the OpNom dictionary; however, in most cases, a name (e.g., of an item or a location) from one data set does not match another from the other data set when these two names point to one single concept or instance. Therefore, a data cleansing task needs to be executed in order to properly populate the ontology.

We are aware of the fact that most of the names are not written in a natural way. Therefore, classical string matching

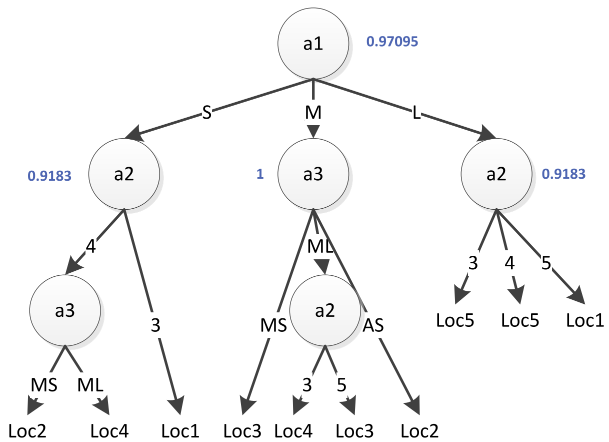


Fig. 5. Decision tree constructed from the training data in Table III.

algorithms in the field of natural language processing are inapplicable. Instead, we use the following algorithm to discover similarities between names.

Given two names name_1 and name_2 where name_1 is a set containing an ordered list of characters $c_1, c_2, \ldots, c_{n_1}$ and name_2 contains $c'_1, c'_2, \ldots, c'_{n_2}$.

$\text{Name}_1 = \text{name}_2$ if $c_1 = c'_1, c_2 = c'_2, \ldots, c_{n_1} = c'_{n_2}$ and $n_1 = n_2$. This is a perfect match and an equivalence can be automatically established.

Name_1 is probably equivalent to name_2 if: 1) $c_1 = c'_{j_1}, c_2 = c'_{j_2}, \ldots, c_{n_1} = c'_{j_{n_1}}$, where $1 \leq j_1 < j_2 < \cdots < j_{n_1} < n_2$ and $n_1 < n_2$; or 2) $c_{i_1} = c'_1,\ c_{i_2} = c'_2, \ldots, c_{i_{n_2}} = c'_{n_2}$, where $1 \leq i_1 < i_2 < \cdots < n_1$ and $n_2 < n_1$.

When name_1 is a proper substring of name_2 or vice versa, a decision of establishing an equivalence between the two names is requested from human experts. For instance, "0.5CTB" and "CTB."

This is a simplified string matching algorithm. We also use more sophisticated algorithms for matching the strings. For example, Levenshtein [48] or string edit-distance is used to find similar names.

After a few iterations, we can get decision rules to automatically decide whether two names, one of which is a proper substring of another, are same or not. One of such rules is shown as follows.

Let us denote the set difference between name_1 and name_2 as S_{diff} and the total number of the members are as $|S_{\text{diff}}|$. If S_{diff} contains only numbers or $|S_{\text{diff}}| \leq 2$, then $\text{name}_1 \neq \text{name}_2$. For example, according to this rule, 0.5CTB $\neq$ CTB and "AA Battery" $\neq$ "AAA Battery."

In the next section, we will illustrate our experiments and experimental conclusions.

IV. Empirical Study

There are in total 1862 items or item groups and 244 containers in the sample data from IMS. Amongst them, 48 items or item groups are directly placed in locations (i.e., these items or item groups are without any containers). All the rest are placed in 15 locations.

There are in total 556 procedures in the sample data from ODF and onboard short term plan (OSTP). These procedures take place at 18 locations. During the task of data cleansing, 74 names of items or item groups from these procedures have a perfect match with the ones from IMS. After cleaned the data, the number is increased into 184.

Both data sets are incomplete. For example, most data concerning item sizes are missing. To solve this problem, we have tried the following three methods: 1) manually insert, i.e., gather product specification from online information; 2) automatic discovery, i.e., finding items of same type with provided sizes; and 3) semi-automatically discovery, i.e., using string matching algorithms to find items of similar names with provided sizes and manually establishing the equivalence between the items.

We have implemented a plugin in an existing toolset called semantic decision tables [49] which contains a set of tools for decision support and ontology-based business rules management. Fig. 6 shows our model illustrated in the model tree and data in the data table. In particular, we show the fact type "ItemOrItemGroup is placed in location" (see the fact type containing roles r_1/r'_1 in Fig. 1), and the population of the roles.

When SIMISS sensors sense a contextual update of an item, e.g., it is moved into a new location by a user at a certain time, the data in Fig. 6 will be updated and the update will be logged. Note that this interface is used by knowledge engineers. Onboard operators normally use devices with compact graphical displays (e.g., smart phones or tablets as illustrated later in this section), which show the information of an item and the contextual change information. These devices support operators' manual inputs when the contextual information is not captured correctly.

It is important to remove the noise in the training data set. Otherwise, the decision tree is not accurate, which will lead to poor searching results. This is the reason why SIMISS is based on an ontology, with which all our data comply.

In the experiments, we compare the searching results with the fact whether an item or item group could actually be found or not. If it is found in another location that is neither as suggested nor random, then we consider it as an instance that is either an exception or can be potentially used to update our training data set.

There are two options of updating the training data set. One is to add new attributes. It requests an update of ontology, which leads to ontology evolution and is not in the consideration of our experiments. The second option is to modify constant values (i.e., C_2–C_{10}). For C_1, we decided to use the time span of the whole mission in order to broaden the searching space.

Table IV shows ten parameter sets concerning sizes of items or item groups. The statistical data of categorizing items or item groups in three sizes (S: small, M: medium, and L: large) are also included.

Suppose that the flat screw driver (barcode: 00000899E) is lost. Its diameter and length are 40 mm and 210 mm. If we use the constants in the parameter set 1 from Table IV (i.e., $C_2 = C_3 = C_4 = C_5 = 10$ and $C_6 = C_7 = C_8 = C_9 = 100$);

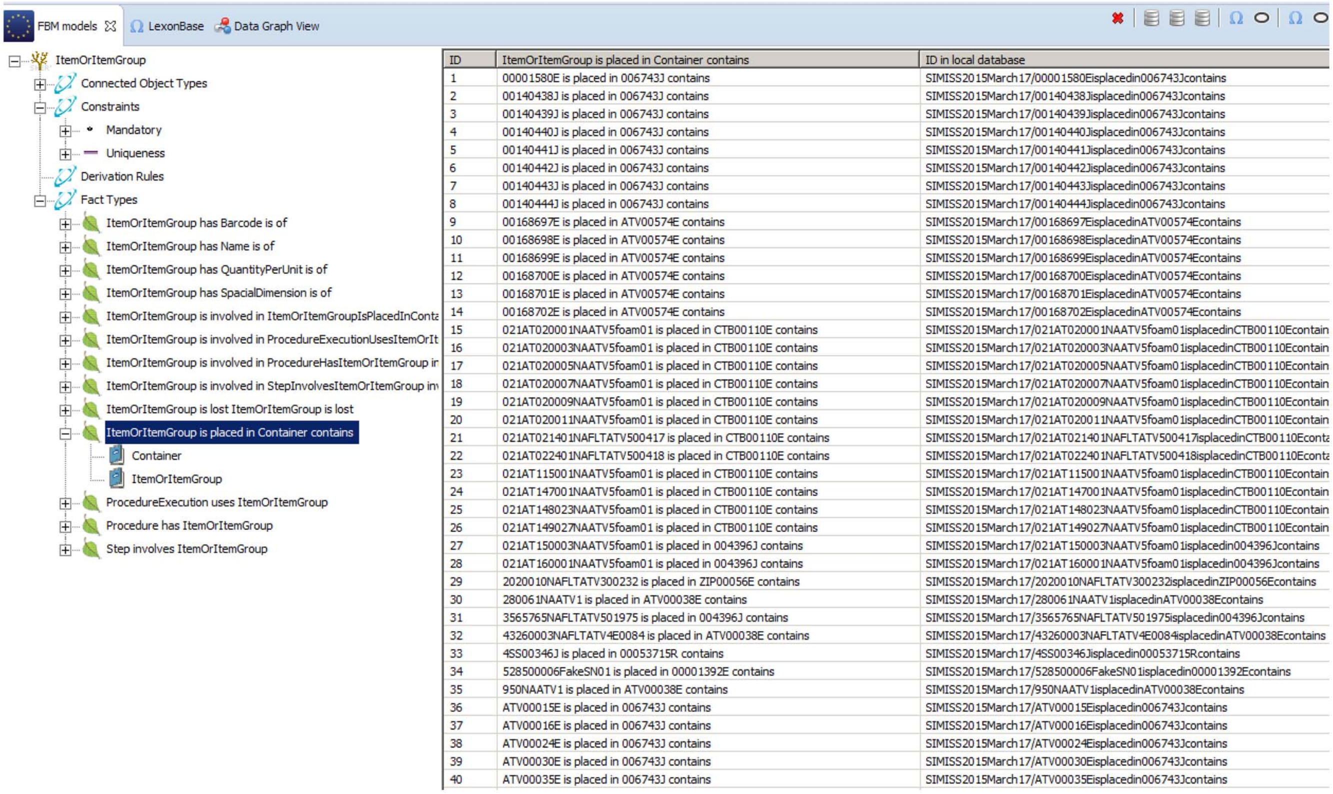


Fig. 6. FBM data viewer.

TABLE IV
PARAMETER VALUES OF $C_1 \sim C_9$; N(S): NUMBER OF SMALL ITEMS OR ITEM GROUPS; N(M): NUMBER OF MEDIUM ITEMS OR ITEM GROUPS; AND N(L): NUMBER OF LARGE ITEM OR ITEM GROUPS

ID	1	2	3	4	5	6	7	8	9	10
c2	10	50	43	20	30	29	10	29	27	100
c3	10	50	39	18	17	29	14	30	35	56
c4	10	50	43	19	20	15	22	20	92	87
c5	10	100	38	46	31	46	54	97	91	41
c6	100	200	91	54	57	95	69	53	366	112
c7	100	200	98	52	76	63	92	76	151	458
c8	100	50	72	78	70	86	73	54	236	172
c9	100	200	61	71	68	601	704	194	347	598
N(S)	1258	1250	1222	1101	1100	1100	1100	1086	1041	1041
N(M)	286	334	265	265	264	336	336	327	540	543
N(L)	320	280	377	498	500	428	428	451	283	280

then it is considered as large (i.e., the value of a_1 is "L"). Given the facts that the procedure with ID "E_BLB_HM_1402" is the last one that uses this item, the experience level of crew one for this procedure is 4 (i.e., the value of a_2 is "4"), and the actual versus expected duration of this procedure executed by Crew One is much longer (i.e., the value of a_3 is "ML").

Seeing that our training data set is quite small, we rebuild the decision tree every time when the training data set is updated. For example, for the experiment using the parameter set 1 in Table IV, we have the training data as illustrated in Table III and the decision tree as illustrated in Fig. 5. When we use the parameter set 5 in Table IV (see the column with ID 5),

TABLE V
UPDATED TRAINING DATA SET: S-SMALL, M-MEDIUM, L-LARGE; AS-ABOUT SAME; ML-MUCH LONGER; AND MS-MUCH SHORTER

ID	a_1	a_2	a_3	Decision
1	S	3	ML	Loc1
2	S	4	MS	Loc2
3	M	3	MS	Loc3
4	M	3	ML	Loc4
5	M	4	AS	Loc5
6	L	5	MS	Loc1
7	M	5	AS	Loc2
8	S	5	ML	Loc3
9	S	4	ML	Loc4
10	L	3	AS	Loc5

we get the training data as illustrated in Table V. The rows with a gray background are the updated ones. The decision tree is rebuilt as shown in Fig. 7.

We have collected a set of items or item groups, the value of a_1 of which vary when we use the parameter set 1 and parameter set 5 in Table IV. Amongst 1862 items or item groups, the values of a_1 of 338 items or item groups are changed.

After feeding the two data sets from these 338 items or item groups to our engine, we get 39 items or item groups that lead to same decisions and 6 that lead to similar decisions.[4]

Fig. 8 shows other statistical results. In this figure, the group of decisions that are modified from Loc_i to Loc_j or vice versa is indicated by loci/j (e.g., loc1/2).

[4]Two similar decisions are a location and its neighbor location (i.e., Loc3 is similar to Loc5 and Loc2 is similar to Loc4).

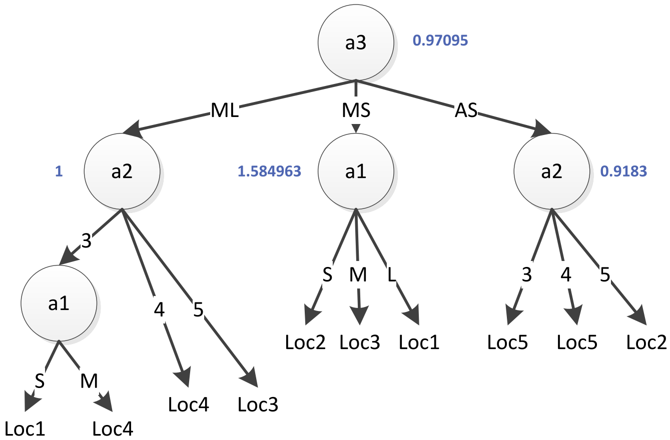


Fig. 7. Decision tree constructed from the training data in Table V.

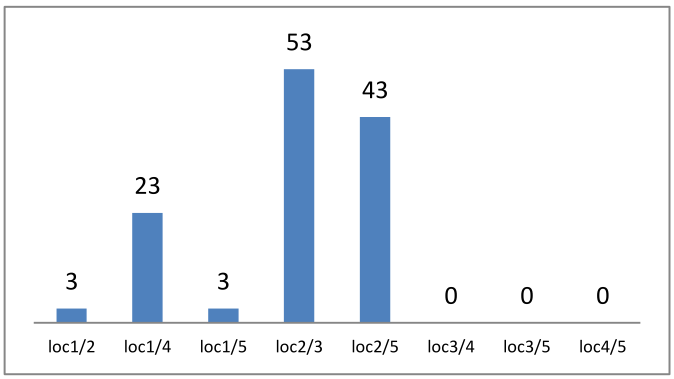


Fig. 8. Modification of decisions between two data sets.

The parameters are set during the test phase. We can still update them during the training phase. During the operation phase, they are not supposed to be updated unless necessary.

For evaluating SIMISS, we use the following measures: accuracy, data sparsity, and time cost. Accuracy is defined as $(\sum_{i=1}^{n} \delta(\mathrm{Loc}_x,)/n)$, where δ is the Kronecker function. A larger accuracy indicates better performance.

Data sparsity is defined as $\sqrt{\sum_{i=1}^{n} (1 - (N_i/N))^2}$, where N_i is the number of the instances of one type, N is the size of the data set in total (i.e., the number of all the instances). A larger sparsity means better performance.

Time cost has two folders. One is the duration from entering a search query until SIMISS gives the answer. The other is the duration from entering a search query until an item is physically found. A lower time cost implies better performance. In our test case, the average time cost of the former was less than a second. It took at least 5 min for the latter; in some cases, it took up to 0.5 day. We thus consider the latter is more important than the former.

Fig. 9 shows how the ubiquitous environment looks like for onboard missions in ISS. An astronaut wares a head-mounted camera, which can be used to scan barcodes of the items involved in a mission. The smart phone around his left wrist can be used to read RFID tags. The tablet in his left hand shows item information.

Fig. 9. ISS onboard simulation (photo credit: ESA).

Fig. 10. Collaboration between the space and the ground segments (photo credit: ESA).

The astronaut on board ISS needs to communicate with the logistics experts on ground. As shown in Fig. 10, operators on the ground, with the help of monitoring systems, support on board crew members when necessary.

We have tested SIMISS using the data from IMS, ODF, and OSTP with a simulation that resides physically outside of the ISS operations center. Our next step is to integrate SIMISS in a demonstration setting as shown in Fig. 9. The context information about the user, location, and time will be captured by (and illustrated on) the smart phone and the tablet. The head-mounted camera will be extended to provide item information to SIMISS (e.g., the ones tagged with RFIDs in the current range). The smart tablet will provide a graphical interface, with which the user can enter a searching query and get information from SIMISS. It will show information more compact than Fig. 6 shows. The tablet will also allow the user to actively participate in the item auditing process.

V. Conclusion

In this paper, we have presented a solution called SIMISS to increase *inventory visibility*, which is one of the most important and fundamental issues in inventory management. On the first step, we formally capture the domain semantics using FBM, with which we generate a data schema for the training set. We use the FBM approach to fulfill this task. On the second step, we select a machine learning algorithm build a decision tree based on this training set. In particular, the ID3 algorithm is applied.

By using SIMISS, we can reduce resupply cost for long duration missions on ISS, reduce "waste" of crew time, and assure mission success.

It can be embedded to wearable devices, which illustrate the searching results in user-friendly formats, e.g., an ordered list of areas, where a lost item can be potentially discovered.

Both items tagged by barcodes and by RFID can be discovered by SIMISS. The only difference is that using RFID is more efficient because SIMISS can immediately update the searching result by replacing an uncertain result with information that is certain when it reaches the readable zone of the RFID tag of a lost item.

SIMISS-based systems can be used during *item auditing*, and during nominal missions operations. Note that the lost items can as well be trashed (i.e., packed into a returning transport vehicle and incarcerated) without notifying the system is also considered untraceable, which is not considered in the use case scenario of this paper.

There is an interesting trend toward the approaches of Semantic Web [50] (also called Web 3.0), IoT [34], and Web of Things [51]. When items from an inventory are physically connected to the Internet, these approaches can be adopted to discover them. Currenlty, due to the infrastructure restriction of IMS, these approaches are not applicable. Nevertheless, we may consider them in the future.

ACKNOWLEDGMENT

The authors are grateful for the help from their colleagues, especially the ESA Columbus logistics engineers B. Bekooy and M. Schenax.

REFERENCES

[1] *Spacecraft Onboard Interface Services–RFID-Based Inventory Management Systems*, CCSDS Secretariat, Washington, DC, USA, 2011.
[2] P. Brown *et al.*, "2E-3 asset tracking on the international space station using global SAW tag RFID technology," in *Proc. Ultrason. Symp.*, New York, NY, USA, 2007, pp. 72–75.
[3] P. W. Fink, R. J. Barton, R. S. Wagner, and P. H. Ngo, "Unified communications for space inventory management," in *Proc. AIAA SPACE Conf. Expo.*, Pasadena, CA, USA, 2009, pp. 1–14.
[4] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquisition*, vol. 5, no. 2, pp. 199–220, Jun. 1993.
[5] N. Guarino, "Formal ontology, conceptual analysis and knowledge representation," *Int. J. Human Comput. Studies*, vol. 43, nos. 5–6, pp. 625–640, Nov./Dec. 1995.
[6] B. J. Angerhofer and M. C. Angelides, "System dynamics modelling in supply chain management: Research review," in *Proc. Simulat. Conf.*, Orlando, FL, USA, 2000, pp. 342–351.
[7] E. Rosch, "Principles of categorization," in *Concepts: Core Readings*. Cambridge, MA, USA: MIT Press, 1999, pp. 189–206.
[8] E. K. Jacob, "Classification and categorization: A difference that makes a difference," *Library Trends*, vol. 52, no. 3, pp. 515–540, 2004.
[9] J. E. Boylan, A. A. Asyntetos, and G. C. Karakostas, "Classification for forecasting and stock control: A case study," *J. Oper. Res. Soc.*, vol. 59, no. 4, pp. 473–481, Oct. 2006.
[10] B. E. Flores, D. L. Olson, and V. K. Dorai, "Management of multicriteria inventory classification," *Math. Comput. Model.*, vol. 16, no. 12, pp. 71–82, 1992.
[11] H. A. Guvenir and E. Erel, "Multicriteria inventory classification using a genetic algorithm," *Eur. J. Oper. Res.*, vol. 105, no. 1, pp. 29–37, 1998.
[12] J. Rezaei and S. Dowlatshahi, "A rule-based multi-criteria approach to inventory classification," *Int. J. Prod. Res.*, vol. 48, no. 23, pp. 7107–7126, Dec. 2010.
[13] C. S. Thachenkary, S. Chatterjee, and J. L. Katz, "Successful product characteristics for electronic commerce: A taxonomy of transaction types," in *Proc. 4th Int. Workshop Community Netw.*, Atlanta, GA, USA, 1997, pp. 77–85.
[14] K.-C. Shang and L.-F. Sun, "Taxonomy in logistics management: A resource-based perspective," *Int. J. Manag.*, vol. 21, no. 2, pp. 149–165, Jun. 2004.
[15] F. Li, Y. Feng, C. Wang, and X. Yang, "Research on product information organization based on ontology in the network," in *Proc. 4th Int. Conf. Wireless Commun. Netw. Mobile Comput.*, Dalian, China, 2008, pp. 1–5.
[16] X. Jin, "Ontology-driven coordination for supply chain system," in *Proc. Int. Conf. Nat. Lang. Process. Knowl. Eng.*, Wuhan, China, 2005, pp. 331–335.
[17] *Fact-Based Modelling Metamodel: Exchanging Fact-Based Conceptual Data Models*, FBM Working Group, Amsterdam, The Netherlands, 2014.
[18] S. G. Nijssen and T. A. Halpin, Conceptual schema and relational database design: A fact oriented approach, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1989, p. 356.
[19] M. Nijssen and I. Lemmens, "Verbalization for business rules and two flavors of verbalization for fact examples," in *On the Move to Meaningful Internet Systems: OTM 2008 Workshops*. Heidelberg, Germany: Springer, 2008.
[20] T. Halpin and T. Morgan, *Information Modeling and Relational Databases*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2008, p. 976.
[21] G. Bakema, J. P. Zwart, and H. van der Lek, *Volledig Communicatiegeoriënteerde Informatiemodellering*. Amsterdam, The Netherlands: Academic Service, 2002, p. 250.
[22] R. A. Meersman, "Semantic ontology tools in IS design," in *Proc. ISMIS Conf.*, London, U.K., 1999, pp. 30–45.
[23] P. Spyns, Y. Tang, and R. Meersman, "An ontology engineering methodology for DOGMA," *J. Appl. Ontol.*, vol. 3, nos. 1–2, pp. 13–39, 2008.
[24] D. A. Jardine, "Concepts and terminology for the conceptual schema and the information base," *Comput. Stand.*, vol. 3, no. 1, pp. 3–17, 1984.
[25] G. Zhao and R. Meersman, "Architecting ontology for scalability and versatility," in *On the Move to Meaningful Internet Systems 2005: Coopis, DOA, and ODBASE*. Heidelberg, Germany: Springer, 2005.
[26] A. Scherp, C. Saathoff, T. Franz, and S. Staab, "Desinging core ontologies," *Appl. Ontol.*, vol. 6, no. 3, pp. 177–221, 2011.
[27] C. Roussey, F. Pinet, M. A. Kang, and O. Corcho, "An introduction to ontologies and ontology engineering," in *Ontologies in Urban Development Projects*, G. Falquet, C. Métral, J. Teller, and C. Tweed, Eds. London, U.K.: Springer, 2011, pp. 9–38.
[28] Y. Tang and R. Meersman, "DIY-CDR: An ontology-based, do-it-yourself component discoverer and recommender," *J. Pers. Ubiquitous Comput.*, vol. 16, no. 5, pp. 581–595, Jun. 2011.
[29] Q. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[30] A. K. Dey, "Providing architectural support for building context-aware applications," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 2000.
[31] H. Attiya and J. Welch, *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, 2nd ed., A. Y. Zomaya, Ed. Hoboken, NJ, USA: Wiley, 2004, p. 432.
[32] M. Weiser, "Some computer science issues in ubiquitous computing," *Commun. ACM*, vol. 36, no. 7, pp. 75–84, Jul. 1993.
[33] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, Aug. 2001.
[34] J. Holler *et al.*, *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*, 1st ed. Amsterdam, The Netherlands: Academic, 2014.
[35] J. Demey, M. Jarrar, and R. Meersman, "A conceptual markup language that supports interoperability between business rule modeling systems," in *Proc. OTM COOPIS DOA ODBASE*, Irvine, CA, USA, 2002, pp. 19–35.
[36] M. Jarrar, "Towards methodological principles for ontology engineering," Ph.D. dissertation, Vrije Univ. at Brussel, Brussels, Belgium, 2005.
[37] K. Angelov and R. Enache, "Typeful ontologies with direct multilingual verbalization," in *Proc. 2nd Int. Workshop Controlled Nat. Lang. (CNL)*, Marettimo, Italy, 2012, pp. 1–20.

[38] Y. T. Demey, "Adapting the fact-based modeling approach in requirement engineering," in *Proc. OTM Workshops*, Amantea, Italy, 2014, pp. 65–69.

[39] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2010, p. 624.

[40] S. Rudolph, "Foundations of description logics," in *Reasoning Web–Semantic Technologies for the Web of Data*, A. Polleres *et al.*, Eds. Heidelberg, Germany: Springer, 2011, pp. 76–136.

[41] M. G. Stewart, "Simulation of human error in reinforced concrete design," *Res. Eng. Design*, vol. 4, no. 1, pp. 51–60, 1992.

[42] R. R. Panko and R. H. Sprague, "Hitting the wall: Errors in developing and code inspecting a 'simple' spreadsheet model," *Decis. Support Syst.*, vol. 22, no. 4, pp. 337–353, 1998.

[43] M. R. Endsley, R. Hoffman, D. Kaber, and E. Roth, "Cognitive engineering and decision making: An overview and future course," *J. Cogn. Eng. Decis. Making*, vol. 1, no. 1, pp. 1–21, Mar. 2007.

[44] C. E. Zsambok and C. A. Klein, *Naturalistic Decision Making*. Hoboken, NJ, USA: Psychol. Press, 2014.

[45] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[46] J. R. Quinlan, *C4.5:Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[47] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1st ed. London, U.K.: Chapman & Hall, 1984, p. 368.

[48] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Phys. Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

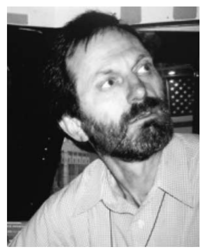
[49] Y. Tang, "On semantic decision tables," Ph.D. dissertation, Dept. Comput. Sci., Vrije Univ. at Brussel, Brussels, Belgium, 2009.

[50] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Sci. Amer.*, vol. 284, no. 5, pp. 34–43, 2001.

[51] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, "From the Internet of Things to the Web of Things: Resource-oriented architecture and best practices," in *Architecting the Internet of Things*. Heidelberg, Germany: Springer, 2011, pp. 97–129.

Yan Tang Demey received the bachelor's degree in computer science and technology from the Northwestern Polytechnical University of China, Xi'an, China, in 2000, the master's and postgraduation degrees in applied computer science from Vrije Universiteit Brussels (VUB), Brussels, Belgium, the joint master's degree in business information management from Solvay Business School, Brussels, and VUB in 2003, and the Ph.D. degree in computer science from VUB in 2009. Her thesis was entitled "Semantic Decision Tables."

She was a Research Fellow with European Space Agency, Noordwijk, The Netherlands, from 2013 to 2015. She is currently a Requirement Engineer and a Search and Knowledge Specialist with the European Patent Office, Rijswijk, The Netherlands. She possesses over ten years of research experience in the fields of ontology engineering, information modeling, and requirements engineering. Since 2004, she has authored or co-authored about 50 scientific papers.



Mikael Wolff received the master's degree in applied physics from the Royal Institute of Technology, Stockholm, Sweden, in 1983.

He possesses over 25 years of experience with software for use by astronauts on board the International Space Station. He holds a Staff position with European Space Agency (ESA), Noordwijk, The Netherlands, in its Engineering Directorate. In his long-term assignment to the ESA Human Space Flight Directorate his research has focused on technologies and implementation for improvement of interactive crew procedures and on board viewers and smart assistance applications for increased performance and satisfaction for the on board astronaut. Providing support for effective on board inventory management is a central tenet of this effort that continues to interest him with its joint cognitive engineering challenges. He was instrumental in migrating the implementation of those systems to commercial mobile devices, and establishing shared real time work spaces between the on board crew member and the ground based group of experts in order to optimize mission return. He has co-authored several papers for space engineering and operations conferences over the last ten years.