Methodology to design management accounting information systems

S V Mkrtychev[1]

[1] Togliatti State University, 14 Belorusskaya St., Togliatti, 445020 Russia

Abstract. The paper considers a methodology to design management accounting information systems (MAIS) based on the object-structured approach, which is an integration of ontological, automata and object-oriented approaches. The phases and stages of the proposed methodology are described. The UML design patterns utilized in this methodology are created drawing on technological ontology classes formalized with the use of the automata approach. This ensures the easy adaptation of MAIS to the specifics of managerial accounting and to the business processes workflow of a particular organization. The main advantage of this methodology is a possibility to implement a management strategy for organization's operational activities already at the stage of designing corresponding problem-oriented control systems through the development of a design patterns set with specified properties. The use of MAIS designed on the base of proposed methodology at the enterprises of textile and woodworking industries provides high efficiency of multi-stage manufacturing control.

CEUR-WS.org/Vol-2258/paper3.pdf

1. Introduction

As it is known, the main functions of Management Accounting Information System (MAIS) are the collection, aggregation, classification and presentation of accounting information in a convenient form for enterprise managers to control costs and to make right decisions.

MAIS belongs to a class of OLTP-systems (Online Transaction Processing) that are implemented in the client-server architecture and utilize a relational database with a strong normalization [1].

At the same time, MAIS is specialized component of the Enterprise Information System (EIS) and can be considered as a key part of problem-oriented control system for operational activities in socio-economic organizations including those operating under uncertainty (complex multi-stage production systems, insurance companies, etc.) [2, 3].

The effectiveness of such systems, in addition to the reliability of output accounting information, depends on the level of compliance of their software architecture with the specifics of managerial accounting not only in a particular organization, but also in a particular country or region [4, 5].

Thus, the research study of methodological aspects of the MAIS design is of particular interest.

2. Background

In modern practice of the MAIS design, at the stage of conceptual modeling, preference is given to the Business Process Modeling (BPM) methodologies [6]. The result of this modeling process is a conceptual model that is a structural-functional (informal) description of MAIS.

It should be noted that for a conceptual representation of MAIS, in addition to an informative description, it is necessary to use more formalized models, for the creation of which it is recommended to utilize methodologies based on Petri networks [7].

However, these methodologies do not allow to fully reveal the features of managerial accounting in complex production systems, since “the focus is on a clear and unambiguous specification of the process and not on particular analysis technique” [8].

To solve this problem, it is recommended to utilize UML design patterns at the stage of MAIS logical modeling [9].

As practice shows, preference should be given to native patterns reflecting the abstractions of the studied subject area and connections between them, for identification of which modeling methodologies based on the ontological approach are used [10].

Thus, for the modeling of accounting information systems, the “Resources - Events - Agents” (REA) ontology is used, which describes the accounting system as a virtual representation of the real business process [11, 12].

The advantages of the REA model include the easy conversion to a conceptual model of a MAIS relational database (entity-relationship model), which makes this ontology successful for modeling business processes in modern ERP-systems [13, 14].

However, despite the fact that the philosophy of REA draws on the idea of reusable design patterns, there is a problem of identification and formalization of the objects represented by REA concepts within the scope of the subject domain [15].

This limits the possibility of utilizing the REA ontology in modeling of MAIS for enterprises with a strong specificity of manufacturing process.

3. Solution approach

To solve the described problem, I propose to utilize an object-structured approach to design MAIS.

In [16] an object-structured analysis methodology is considered, the principles of which are based on an ontology-driven approach representing the enhancement of a classical structured analysis methodology for knowledge engineering.

In [17] an approach to configuring MAIS for multi-stage production is described. This approach is based on the presentation of a multi-stage production system in the form of an object-structured model, the elements of which are virtual heirs to the following basic classes of the technological ontology:

- Class “Aggregate”, the objects of which change the state of a material flow element (raw materials, production, document, et al.).
- Class “Warehouse”, the objects of which store material flow elements and register their movement within a production process.
- Class “Controller”, the objects of which monitor the status of a material flow element and control its movement within a production process.
- Class “Stage”, the objects of which are combinations of the above-described classes (for example, "Warehouse-Aggregate-Warehouse").

The rules for the above object classes behavior are determined by transaction axioms of the enterprise ontology [18] and the rules for conducting managerial accounting in a particular organization.

Business processes are managed based on an event-driven principle with the adjustment to changes in the status of the processing element of material or information flow.

This model can be presented in the form of a directed graph and can be described quite simply using an incidence matrix or an array [19].

The physical implementation of the object-structured model of MAIS is a transaction of the OLTP-system that provides information processing during the production process.

The advantage of MAIS object-structured models is the universality that is provided through their isomorphism.

Taking into account the well-known propositions of isomorphism and invariants of directed graphs [20], the following statement of object-structured models isomorphism is proposed: the object-structured models of MAIS are isomorphic if the graphs of the models are isomorphic, and there exists a bijection between the ontological classes, the heirs to which are objects denoting nodes and arcs of object-structured models graphs.

Thus, isomorphism verification for object-structured models of MAIS for similar business processes is reduced to comparing the objects properties that are designated by the corresponding elements of the graphs of the compared models as to their belonging to the same class of the technological ontology.

The concept of the object-structured approach proposed in this article is based on the integration of ontological, automata and object-oriented approaches and seems more promising for modeling of problem-oriented MAIS.

4. Design methodology

The MAIS design methodology consists of the following phases.

4.1. Object-structured modeling

Figure 1 shows an object-structured model for the system of *N*-stage processing of accounting information.

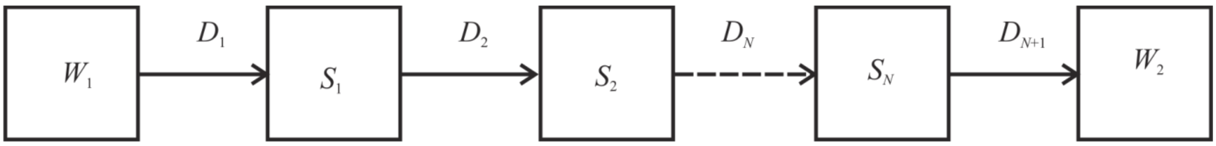

Figure 1. An object-structured model for the system of *N*-stage processing of accounting information.

This object-structured model is described as the directed graph $O(W, S, D)$, where:

- $W = \{W_1, W_2\}$ – nodes denoting the objects of the "Warehouse" class.
- $S = \{S_1, S_2,..., S_N\}$ – a set of nodes denoting the objects of the "Stage" class.
- $D = \{D_1, D_2,..., D_{N+1}\}$ – a set of arcs loaded with the elements of information flow.

The mathematical description of such a model can be represented as ordered arrays of the following form:

$$MW : \text{array } [1..2] \text{ of } CW, \; MS : \text{array } [1..N] \text{ of } CS \text{ and } MD : \text{array } [1..N+1] \text{ of } CD, \text{ where:}$$

CW, *CS*, *CD* – the data types that determine the subsets of the indicators values, with which the nodes or arcs of the graph are loaded, respectively (for example, item balances, statuses or movements of goods or documents in warehouses and stages of the technological process).

Array element indices are the numbers of nodes or arcs in a graph.

4.2. Formalization of the elements of the object-structured model

To formalize the elements of the object-structured model of MAIS, an automata approach is used [21].

This approach is a variation of a process-oriented approach and is a representation of the object-structured model of MAIS as a system of interacting automata controlling the status of the processed element of the material (information) flow, which is described as a finite-state automaton in accordance with its life cycle.

Table 1 presents an example of a product life cycle in a multi-stage production system.

Table 1. Product life cycle.

Status	Description
1	Raw material
2	Semi-finished product
3	Finished product

It should be noted that the automaton approach is not related to the specification of the UML notation, which significantly limits its capabilities for creating design patterns and logical modeling of MAIS.

In [22] a method for designing operations of state-dependent object classes with the use of a UML statechart diagram is proposed.

The essence of this method can be described as follows: at the stage of designing a class based on a finite-state automaton, the actions of the statechart diagram of the automaton, as a rule, are mapped onto operations of the designed class.

Thus, the statechart diagram of a finite automaton shows the actions and activities initiated as a result of the transition of its states. In this case, it is meant that the actions, as a rule, are displayed on the operation of the class.

However, this method does not answer the question of how the properties of ontological classes are related throughout the automata representations with the properties of the object classes created on their basis.

To solve this problem, in the proposed design methodology, object representations for the elements of the object-structured model are created using the transformation of the automaton set-theoretic description into the specifications of a UML class diagram.

Let *R* be an automaton formalizing an element of the object-structured model of a MAIS.

Then, the transformation of the set-theoretic description of the automaton *R* to the object model *M* of the object-structured model MAIS element can be described as follows:

$$M\ (N_{CO}, S_A, S_O) \rightarrow \Psi(R(N_{TO}, A_{TO}, V_R)), \text{ where:}$$

- N_{TO} - the name of the technological ontology class, the heir to which is an element of the object-structured model.
- A_{TO} – a set of attributes of the technological ontology class describing the state of the automaton *R* (shown in Table 2).
- $V_R = (N_v, L_v)$ – functionality of the automaton *R*, where:
  N_v – names of automata functions.
  L_v – the descriptions of algorithms that realize the automata functions, examples of which are presented in Table 3.
- Ψ – an operator transforming the elements of the automaton set-theoretic description into an object model (presented in Table 4).

Table 2. Attributes of technological ontology classes.

Class name	Attribute
Warehouse	Item balance
Aggregate	Item status
Controller	Item status control result

Table 3. Descriptions of automata functions.

Automaton	Function name	Transition algorithm
Warehouse	Receipt/Expense the Item	Recalculating the item balance
Aggregate	Change the Item Status	Changing the status of the item
Controller	Control the Item Status	Controlling the status of the item

Table 4. Transforming an automaton set-theoretic description into an object model.

Set-theoretic description element	Object model element
Name of technological ontology class, N_{TO}	Name of object class, N_{CO}
State description, A_{TO}	Attribute specification, S_A
Function description, V_R	Operation specification (signature and method), S_O

4.3. Development of UML design patterns

At this phase, the object models of the same type elements of the MAIS object-structured model are integrated into groups with common classes of the technological ontology, based on which the design patterns in the UML notation are created.

The created patterns, at the stage of logical modeling, are utilized as superclasses to build inheritance object models that are used to develop a relational data model and business logic of MAIS.

Figure 2 illustrates design patterns of MAIS for multi-stage manufacturing.

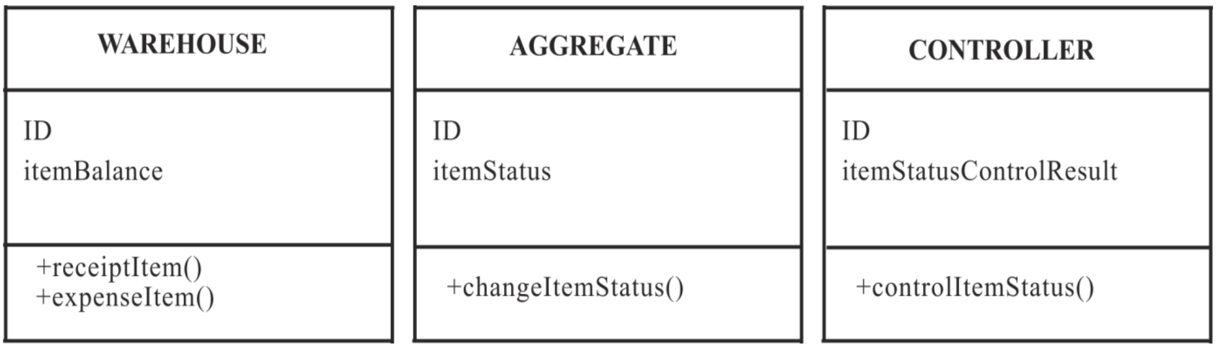

Figure 2. Design patterns of MAIS for multi-stage manufacturing.

It should be noted that the utilization of the superclass concept at the stage of logical modeling of the MAIS provides a stable model for the inheritance of the objects involved in the system - heirs to the corresponding ontological class.

Preference should be given to the superclasses, on the basis of which ready-made design patterns will be created [23].

It is also very important to pay attention to the principle of function interoperability for operations of the selected object group and the used superclass.

Otherwise, you should create both a new superclass, and a design pattern based on it.

5. Advantages of the proposed methodology

Advantages of the proposed methodology are:

5.1. Easy adaptation of MAIS object-structured models to the specifics of managerial accounting in a particular organization.

This advantage is achieved due to the possibility of adapting the object-structured approach to the features of the subject domain by modifying basic ontological classes or creating new ones.

Thus, drawing on basic ontological classes of the technical ontology, ontological classes for the MAIS design for operational insurance activities were developed [24]:

- “Insurance Document” - an active document involved in the operational business process (insurance policy, claim form, et al.).
- “Insurance Operator” – a subclass of “Aggregate” representing a person involved in the processing of insurance documents (agent, claims adjuster).
- “Insurance Inspector” – a subclass of “Controller” representing a person who provides control of insurance documents or the identification of potential risks and decision-making on whether to accept or reject an application (underwriter, claims inspector).
- “Insurance Portfolio” – a subclass of “Warehouse” representing a repository of insurance documents (portfolio of the insured, portfolio of the insurer).

Figure 3 shows examples of design patterns models and MAIS objects inheritance models for operational insurance activities.

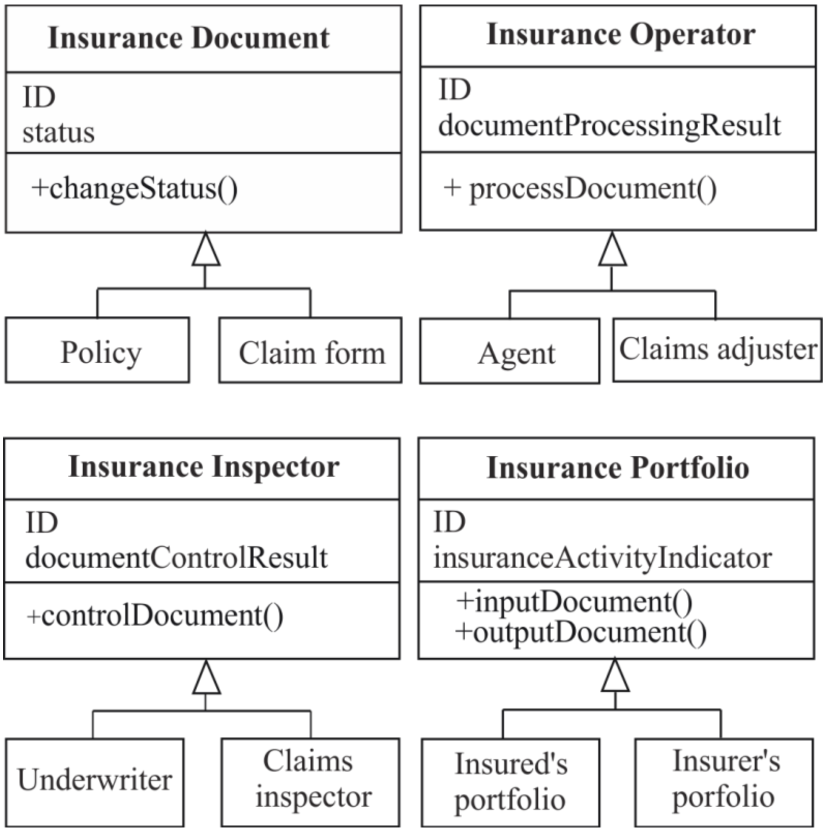

Figure 3. Design patterns and MAIS objects inheritance models for operational insurance activities.

It should also be noted that the use of polymorphic operations in design patterns provides flexibility of reconfiguring the properties of inherited objects and, consequently, the easy adaptation of object-structured models to the specifics of the managerial accounting of a particular organization.

5.2. Easy integration of MAIS into EIS of an organization.

It is important to note that MAIS objects, as a specialized component of the organization's EIS, should be present in their representation also as a database and business logic objects (for example, in the EIS of manufacturing enterprises these are directories of warehouses, aggregates, routes of technological processes, etc.; in the EIS of insurance companies - directories of insurance agents, customers, data accumulation registers, etc.).

Thus, at the stage of physical modeling, the ease of the MAIS integration is achieved by the building of its server modules on the EIS platform, for example, by adding the objects and connections to its database structure, which are needed to support the required functionality and to ensure strong database normalization.

6. Conclusion

The paper considers a design methodology for MAIS based on an object-structured approach.

The integration of ontological, automata and object-oriented approaches within the object-structured approach allows us to expand its methodological possibilities to design effective MAIS.

The main advantage of this methodology is a possibility to implement a management strategy for organization's operational activities already at the stage of designing corresponding problem-oriented control systems through the development of a design patterns set with specified properties.

Due to universality of the object-structured model used in the approach, MAIS is easily adapted to the specifics of managerial accounting in a particular organization and to the integration into its EIS.

Thus, based on the proposed methodology, a configuration of MAIS for multi-stage manufacturing on the platform "1C: Enterprise 8.x" has been implemented. Utilizing virtual warehouses in MAIS as real storage locations allowed us to increase the accuracy of calculations for normative production losses and to organize managerial accounting thereof with the reference to individual stages of the technological process. The use of this MAIS at the enterprises of textile and woodworking industries provided high efficiency of multi-stage manufacturing control.

The proposed methodology can be used to design MAIS in socio-economic organizations, for instance, for multi-stage production systems and insurance activities.

7. References

[1] Bog A, Sachs K, Zeier A and Plattner H 2011 *Normalization in a Mixed OLTP and OLAP Workload Scenario* 3rd TPC Technology Conference on Performance Evaluation & Benchmarking

[2] Rom A and Rohde C 2007 Management accounting and integrated information systems: A literature review *Int. J. of Acc. Inf. Sys.* **8** pp 40-68

[3] Alikhani H, Ahmadi N and Mehravar M 2013 Accounting information system versus management information system *European Online J. of Nat. and Soc. Sci.* **2**(3) pp 359-366

[4] Mia L and Chenhall R 1994 The usefulness of management accounting systems, functional differentiation and managerial effectiveness *Acc. Org. and Soc.* **19**(1) pp 1-13

[5] Choe J and Langfield-Smith K 2004 The effects of national culture on the design of management accounting information systems *J. of Comp. Int. Man.* **7** pp 3-24

[6] Panayiotou N and Tatsiopoulos I 2013 *Supporting the design of a management accounting system of a company operating in the gas industry with business process modeling* Advances in production management systems, competitive manufacturing for innovative products and services (Berlin: Springer) pp 686–692

[7] Kim R, Gangolly J and Elsas P 2017 A framework for analytics and simulation of accounting information systems: A Petri net modeling primer *Int. J. of Acc. Inf. Sys.* **27** pp 30-54

[8] Van der Aalst W 2009 *Transactions on Petri Nets and Other Models of Concurrency II* Special Issue on Concurrency in Process-Aware Information Systems (Eindhoven : Technical University) p 297

[9] Chen H and Morris B 2009 Using Design Patterns, Analysis Pattern, and Case-Based Reasoning to Improve Information Modeling and Method Engineering in Systems Development *Int. J. of App. Man. and Technol.* **7**(1) pp 19-49

[10] Hammar K 2017 *Content Ontology Design Patterns: Qualities, Methods, and Tools* vol 1879 (Linköping: University Electronic Press)

[11] McCarthy W 1982 The REA accounting model: A generalized framework for accounting systems in a shared data environment *The Acc. Rev.* **57** pp 554-578

[12] Geerts G and McCarthy W 2002 An ontological analysis of the economic primitives of the extended-REA enterprise information architecture *Int. J. of Acc. Inf. Sys.* **3** (1) pp 1–16.
[13] Romney M and Steinbart P 2009 *Accounting information system* 11th ed Pearson International Editions
[14] Fallon R and Polovina S 2013 REA Analysis of SAP HCM; Some Initial Findings *Proc. of the 3rd CUBIST Workshop* p 31
[15] Hrby P 2006 *Model-Driven Design Using Business Patterns* (Berlin: Springer)
[16] Gavrilova T, Kurochkin M and Veremiev V 2004 Teaching Strategies and Ontologies for E-learning *Int. J. Inf. Theor. and App.* **11** (1) pp 35–42
[17] Mkrtychev S 2014 Object-structured approach to modeling of problem-oriented management accounting information systems *Bulletin of the Tomsk Polytechnic University* **325** (5) pp 66 –71
[18] Dietz J 2006 *Enterprise Ontology: Theory and Methodology* (Berlin: Springer)
[19] Mkrtychev S, Ochepovsky A and Enik O 2018 Configuration of management accounting information system for multi-stage manufacturing *J. Phys.: Conf. Ser.* **1015** 042039
[20] Rosen K 2012 *Discrete Mathematics and Its Applications* 7th ed (New York: McGraw-Hill)
[21] Shalyto A and Tukkel N 2001 SWITCH Technology: An Automated Approach to Developing Software for Reactive Systems *Programming and Computer Software* **27**(5) pp 260–276
[22] Gomaa H 2011 *Software modeling and design* (Cambridge : University Press) p 578
[23] Larman C 2004 *Applying UML and patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (New Jersey: Prentice Hall) p 736
[24] Mkrtychev S 2015 Methodology of modeling of problem-oriented management accounting information systems in insurance *Modern problems of science and education* **1**(1) p 174