ISSN 2518-7325 (Online), ISSN 2306-1677 (Print)
Land Management, Cadastre and Land Monitoring
Received: 09.12.2025; Accepted: 25.03.2026; Published: 30.03.2026;
http://dx.doi.org/10.31548/zemleustriy2026.01.09

UDC 004.9:911.5/.9:528.94

MODEL-DRIVEN SOFTWARE ENGINEERING METHODOLOGY IN THE FRAMEWORK APPROACH

O. Dyshlyk, *Executive Director, "Geomatic solutions" LLC,*
E-mail: dyshlyk@geomatica.kiev.ua
ORCID: 0000-0002-8066-3925
V. Chabaniuk, *Ph.D. Physical and Mathematical Sciences,*
Senior Researcher,
E-mail: chab3@i.ua
ORCID: 0000-0002-4731-7895
Institute of Geography of the National Academy of Sciences of Ukraine,

Annotation

The research of the "Framework Approach for Handling (Hierarchical) Spatial Information Systems (SpIS)", hereinafter referred as the Framework Approach, was continued. Prior to this, the entities of the Framework Approach, which relate to the upper echelons of its epistemological hierarchy, were considered. Namely, to (the title of the article corresponding echelon is given in quotes "..."): 1) the first echelon – "Framework Approach as a Strategy for Research and Design of Complex SpIS (on the Example of NGDI)"; 2) again the first echelon – "Microsoft Solutions Framework (MSF) as a Generalized Methodology of the Framework Approach for Handling SpIS"; 3) the second echelon – "Standardization of the MSF Agile Methodology using ISO/IEC 24744 standard"; 4) the third echelon – "Usage

the Methodics of the Framework Approach to Creating SpIS Using Modern Microsoft Technologies". Echelons unite groups of users of a modeling system. They correspond to strata of the same system.

Three of the four mentioned articles make significant use of special knowledge about MSF and about modern information technologies (IT) of Microsoft. To reduce dependence on this knowledge, in this article we used "neutral" knowledge about software development methodologies. For this purpose, the so-called Model-Driven Software Engineering (MDSE) is considered through the "framework prism". The latter correlate with Model-Based System Engineering (MBSysE) and, more generally, with Model-Based Engineering (MBE). Their understanding is necessary to create our own methodology, which is called the Pattern-Based Spatial Engineering (PBSpE) methodology. PBSpE will include the methodics of extending the SpIS, which were mentioned in our works earlier. Moreover, these methodics have already been used by us in practice, sometimes without association with any methodology. It should be noted that our SpIS extension methodics are applied to constructions that are already structured as a result of using the so-called Atlas Solutions Framework AtlasSF.

Keywords: *Model-Driven Software Engineering (MDSE) methodology, MDSE Conceptual Framework, SpIS extension methodics.*

Introduction

The article [1] compares the concepts of "approach" and "methodology", which are close to each other, but methodology is a more practical concept. There, methodology: 1) refers to specific procedures, methods and certain tools used to perform a specific task or research project; 2) is a systematic and structured way of collecting data, analyzing information or solving a problem; 3) is more detailed and practical compared to the approach, outlining the step-by-step process that must be performed to achieve the goals of the approach; 4) includes methods, techniques and tools that will be used to collect and analyze data, test hypotheses or reach conclusions. In dictionaries [2] METHODOLOGY (from the Greek. μετέδοζ –

method, way of research or cognition, λόγοζ – doctrine; together - the doctrine of the method) - A set of approaches, ways, methods, techniques and procedures used in the process of scientific cognition and practical activity to achieve a predetermined goal.

The term "approach" in the cited article understood in three senses: 1) as a strategy for using GeoInformation Systems and Technologies (GIS&T) in the management of large territories (LT) of Ukraine; 2) as a methodology of the Framework Approach for handling SpIS, generalizing the methodology of the Microsoft Solutions Framework (MSF); 3) as a general (γ-) method for studying complex spatial phenomena and designing their models (SpIS) using Conceptual Frameworks of "subject X" and Solutions Frameworks of "subjects XY", where X can take the values of both NGDI, NSII, and other rather arbitrary SpIS. Another change in subject X depends on changes in values of Y, which denote belonging to the stratum. That is, XY used to denote different values of subject X, which may depend on the values of stratum Y.

The methodics of the SpIS extension will be included in the Pattern-Based Spatial Engineering (PBSpE) methodology. SpIS include the National Geospatial Data Infrastructure (NGDI) and the National Spatial Information Infrastructure (NSII). BPSpE is currently being developed by us. It will be a specialization of the Model-Based Systems Engineering (MBSysE) methodology. We expect that a review of the software development methodology from a “neutral” viewpoint will complement the understanding of the MSF Agile methodology [3] and help clarify the methodics of the SpIS extension. The Framework Approach methodics with using modern Microsoft technologies [4] will become clearer when creating a SpIS also as the general (γ-) method of research and design using the Conceptual Frameworks of “subject X” and the Solutions Framework of “subjects XY”.

Relevance and Analysis of recent researches and publications

The relevance of the work stems from the following evolution of the Framework Approach, which has become increasingly important over the years. This is explained both by the extension of the domain and subjects of inquiry, and by the improvement of the quality of modeling solutions. The Framework Approach to the

Research and Design of SpIS began at the turn of the millennium with the Solutions Framework (SoFr) of subjects X, where X acquired the values of specialized SpIS, for example, Electronic Atlases (EA). Over the years, it has been realized that SoFr X[Y] can be interpreted as systemic methods of handling (creation, operation, etc.) of the SpIS within the framework of individual information projects. The subjects of such project are not only the system as a whole, but also its products or subsystems, denoted by XY, where Y could vary depending on the value of X. The second group of systemic methods of the Framework Approach are the methods of Conceptual Frameworks of Subjects X. At the moment, we have already published two articles on the current state of Conceptual Frameworks, and the second article [5] provides formalization.

Before presenting the main material, we would like to recall the Classic Solutions Frameworks (SoFr) Model of subjects X[Y] (Fig.1), which we have been using for the last 25 years. Mainly in the practice of creating various SpIS (subjects X), denoted SpIS SoFr. Among SpIS special attention paid to the “classic” Atlas SpIS: Electronic Atlases (EA) and Atlas Information Systems (AtIS).

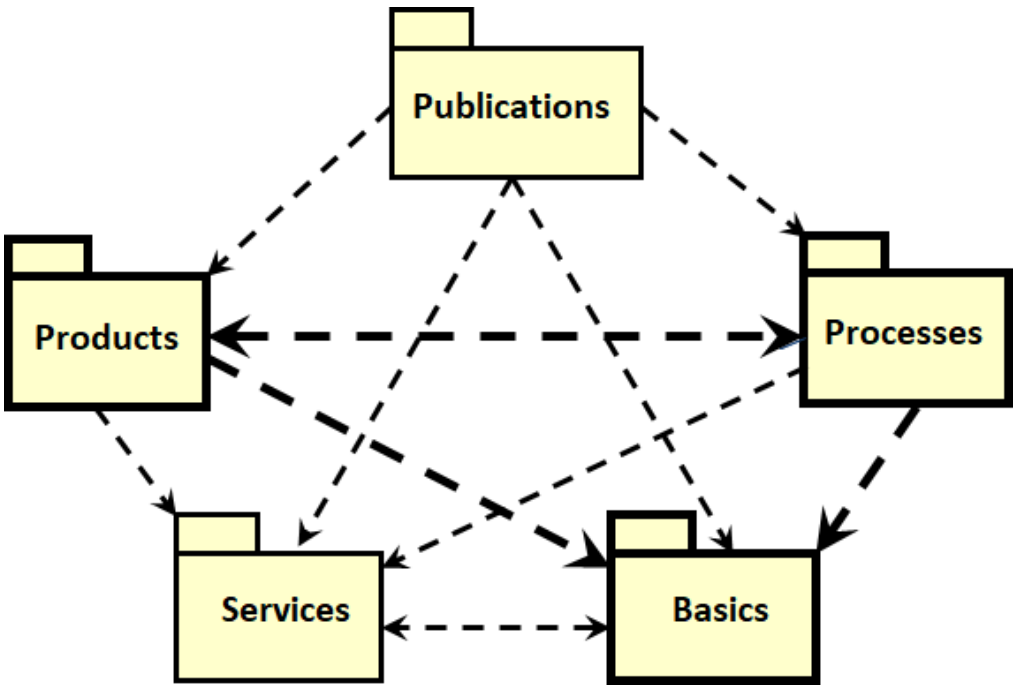

Fig. 1. (Classic) Solutions Frameworks (SoFr) Model of subjects X

The Atlas SpIS SoFr have been significantly used in our recent works. In particular, in the project of the National Research Foundation of Ukraine 2022.01/0121, which carried out at the Institute of Geography [6]. Recently, the evolution of SpIS frameworks has led to the so-called "Framework approach to the research and design of (arbitrary, not only Atlas) SpIS" [1]. If we reduce the understanding of the Framework Approach to a set of homogeneous Framework methods, then Fig. 2 will be valid.

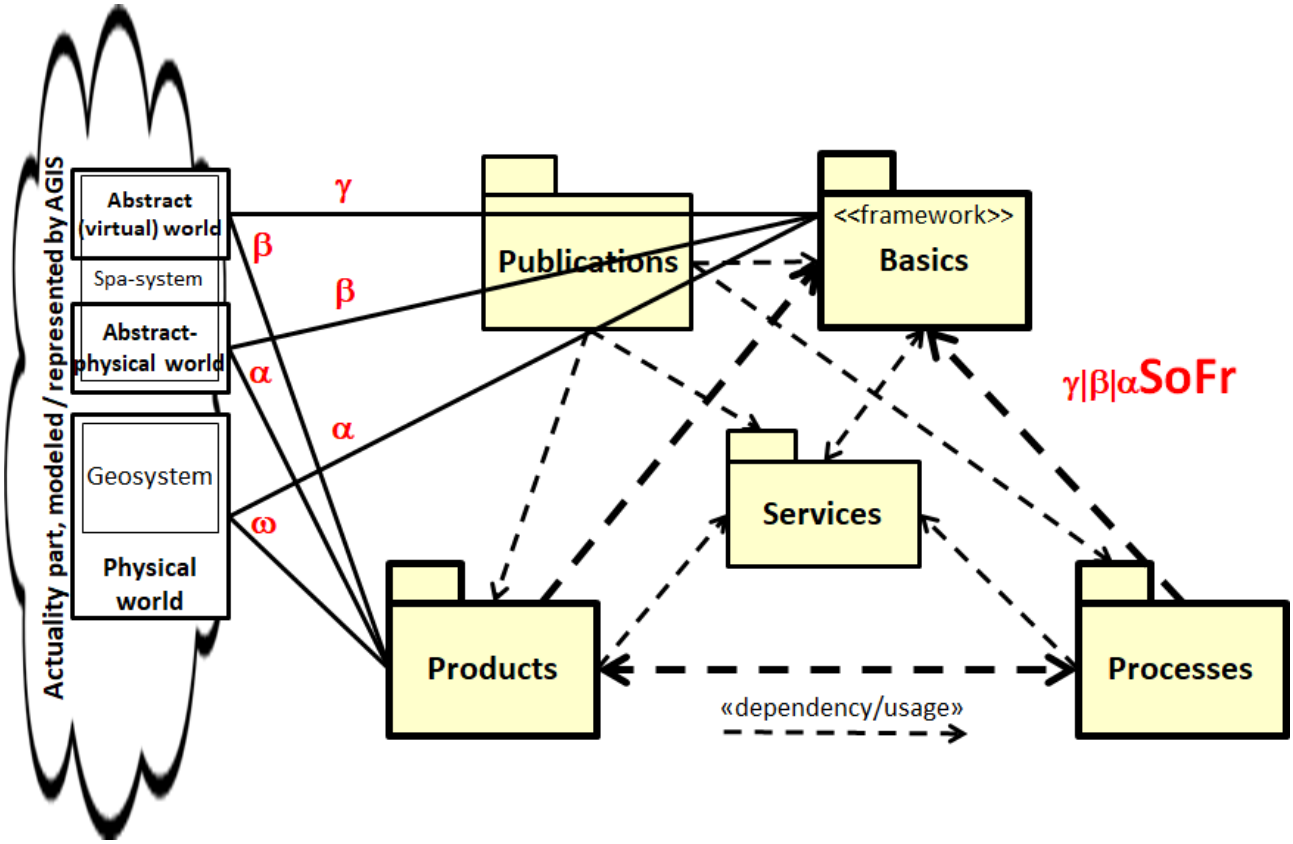

Fig. 2. γ|β|αSoFr as research methods, together – SoFr approach

The right part of Fig. 2 shows a modified petrade of the "classic" SoFr X model (Fig. 1), and is marked with the notation γ|β|αSoFr in red. This means that the same petrade is used to represent γSoFr, βSoFr and αSofr. γ, β, α also denote the hierarchical strata of the modeling SpIS related to the methods (see below): General, Conceptual, Application, and/or the corresponding echelons of user groups. In practice, the stratified resulting SpIS is obtained by applying a certain extension technique. The extension can be performed: 1) "bottom-up", from the Electronic

Atlas or Atlas Information System (EA or AtIS, α) to the GeoInformation Platform (GIP, γ). It called the Atlas Extension (AtEx); 2) "top-down", from GIP to EA/AtIS. It called GeoInformation Extension (GIE); 3) combining 1) and 2) between different strata.

The X SoFr approach is shown as a set of the following methods of actuality inquiry: 1) γ-method - the relation between the Abstract (virtual) world and γ-Basics. γ-Basics include the β-Products and β-Processes patterns. If the context is the creation of EA/AtIS, then γ-Basics also include the pattern of the Base Map, used in EA/AtIS; 2) β-method - two relations: between the Abstract-Physical World (APW) and β-Basics, and between APW and α-Products. The presence of these two relations explained by the existing practice of using two types of modeling: ontological or linguistic; 3) α-method - two relations: between the Physical World (PW) and α-Basics and between PW and ωProducts.

he specification of Fig. 2 is shown in Fig. 3. It should be understood as an explanation of part of the Framework Approach to Research and Design of SpIS. Three types of patterns are used: 1) analytical patterns, or γ-patterns, or patterns of the General Stratum; 2) design patterns, or β-patterns, or patterns of the Conceptual Stratum (like GeoSF); and 3) application patterns, or α-patterns, or patterns of the Operational Stratum (like AtlasSF 1.0). The version number 1.0 of AtlasSF allows to separate αSoFr from SoFr of higher strata. The fact is that AtlasSF can be an approach, a method, and a tool [6]. In this article, αSoFr AtlasF1.0 is consistent with the SpISframeworks (see below).

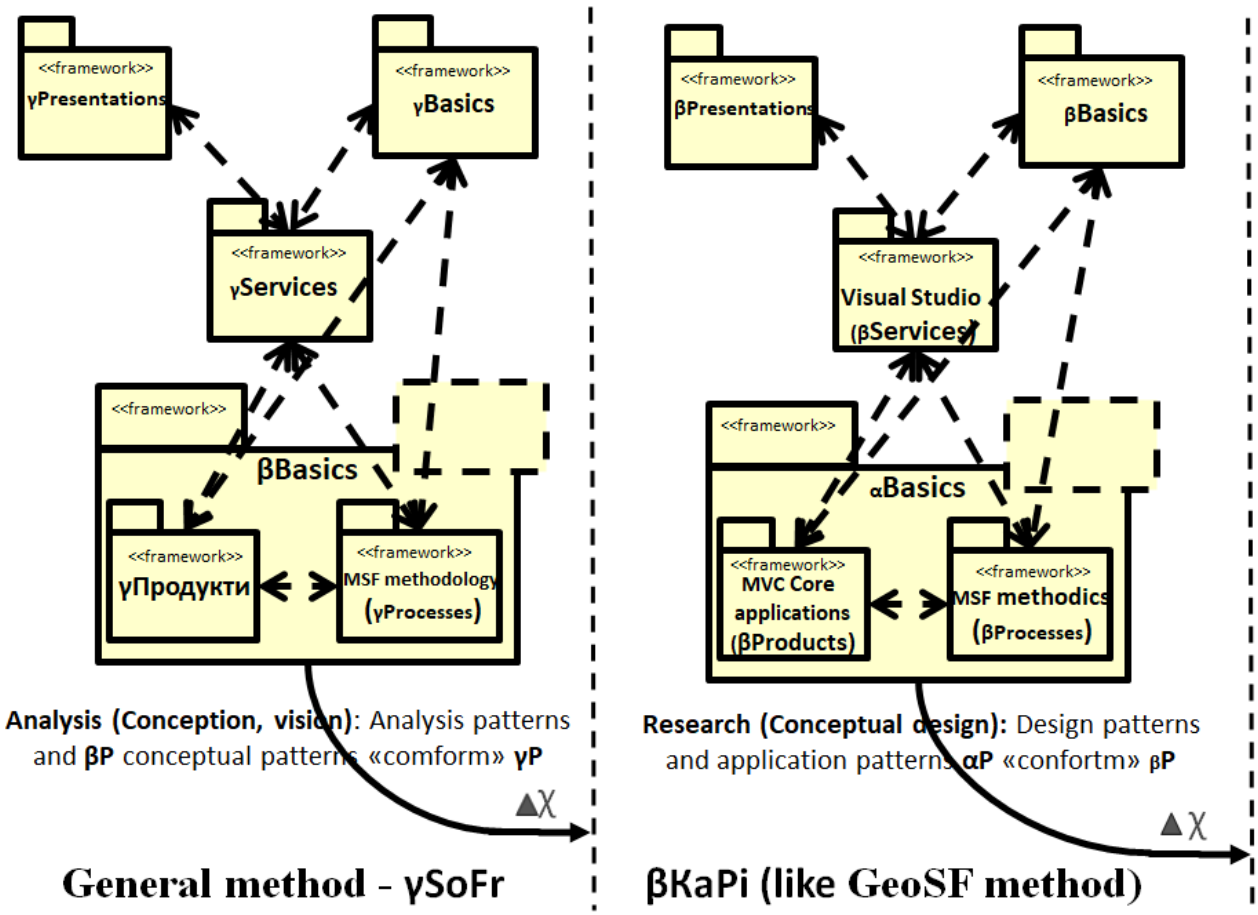

Fig. 3. Examles of γSoFr and βSoFr methods. Use of Visual Studio, MSF methodology, MVC Core applications and MSF methodics described in [4]

To understand the practice of the Framework Approach usage, a hierarchy of specific, rather general SpIS is used, which belong to the set of Atlas Geoinformation Systems (AGIS) of Large Territories (LT) – AGIS LT. AGIS LT includes both all known classic Atlas Systems and, among others, some “non-classic” SpIS.

Our practice has been materializing through information (information-software) systems for quite some time, so we got our first ideas about SoFr X from the practice of implementing information projects, which corresponds to the “bottom-up” expansion method. “Practically” immediately after practical implementation, a “theoretical” idea of patterns began. This was the beginning of the movement in the “top-down” direction, or from theory to practice.

Research purpose

The purpose of the work is to describe one of the methodologies of the "Framework Approach to the Research and/or Design of SpIS". It focuses on a subset of SpIS - software systems, and known as Model-Based Software Engineering (MBSE). Due to the shown correlation with the Conceptual Framework, MBSE can be considered from the viewpoint of its possible use in the Pattern-Based Spatial Engineering (PBSpE) we are creating.

Materials and research methods

The work uses two system framework methods: Conceptual Frameworks (CoFr) and Solutions Frameworks (SoFr) [7]. Practice and methodology understood as components of a hierarchy of interdependent concepts, as shown in Fig. 4. There compared to its original in the article [1]: 1) the caption has been changed, although the correct one would also be "An example of usage the CoFr system method to AGIS-LT"; 2) Method/Practice/Technology/SpISframeworks now indicates three echelons (in the original only the Operational echelon indicated): 2.1) External Operational echelon, 2.2) Operational echelon, 2.3) Application echelon; 3) SpISframeworks has been added to Method/Practice/Technology.

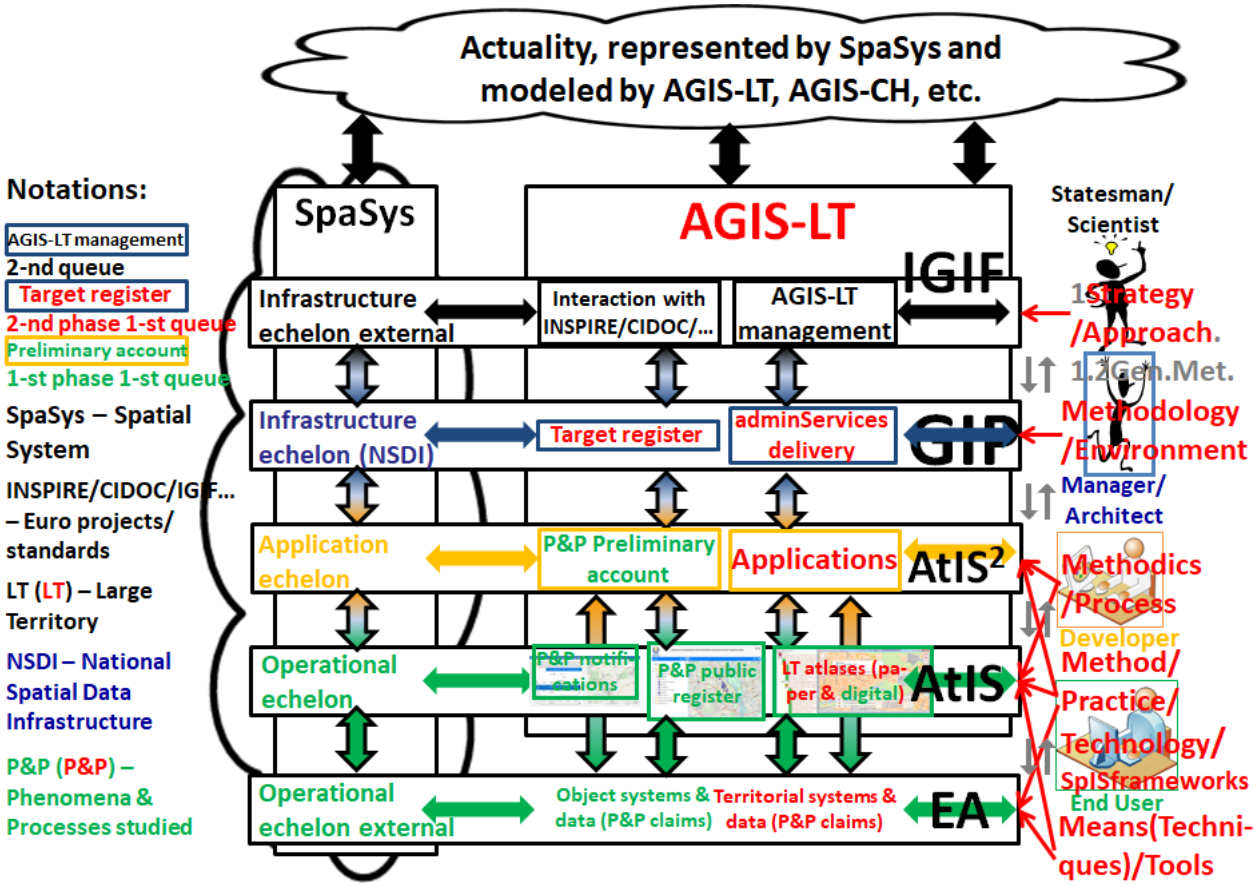

Fig. 4. Hierarchy of AGIS strata and echelons using the AGIS-LT example

Both systems framework methods are used as shown in the diagram Fig. 5 for an arbitrary pattern in UML notation. The rectangles shown with a dotted line can be used to indicate the initial values of the patterns.

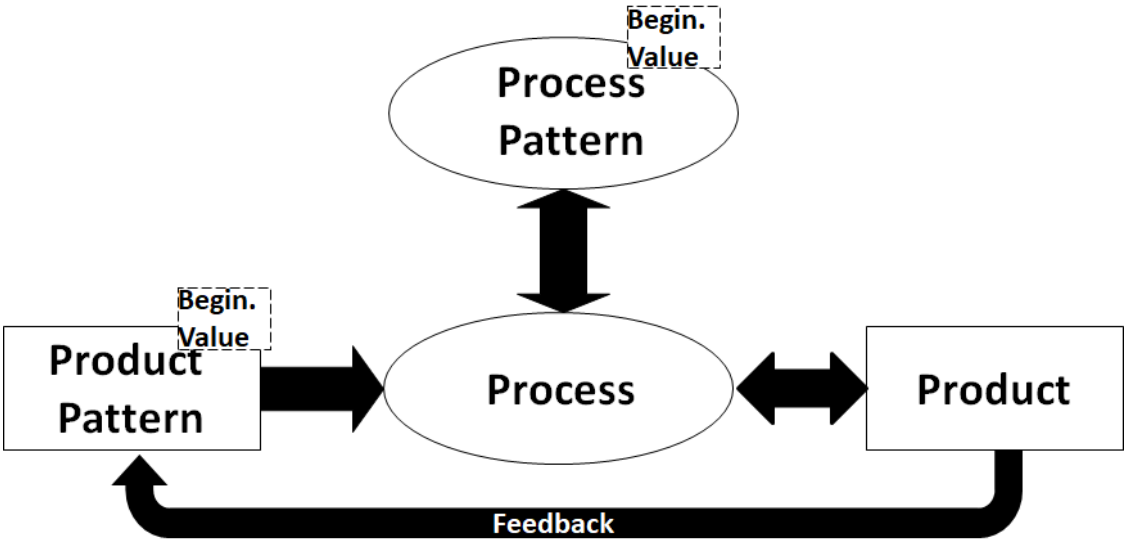

Fig. 5. Pattern usage scheme

For example, in the articles [8], the scheme Fig. 5 is shown as in Fig. 6. We draw your attention to the fact that the "product" pattern of AGIS (product pattern) is the Atlas GeoInformation Model (AGIM), the initial value of which is shown (in the dotted rectangle) as the Electronic Version of the National Atlas of Ukraine (ElNAU).

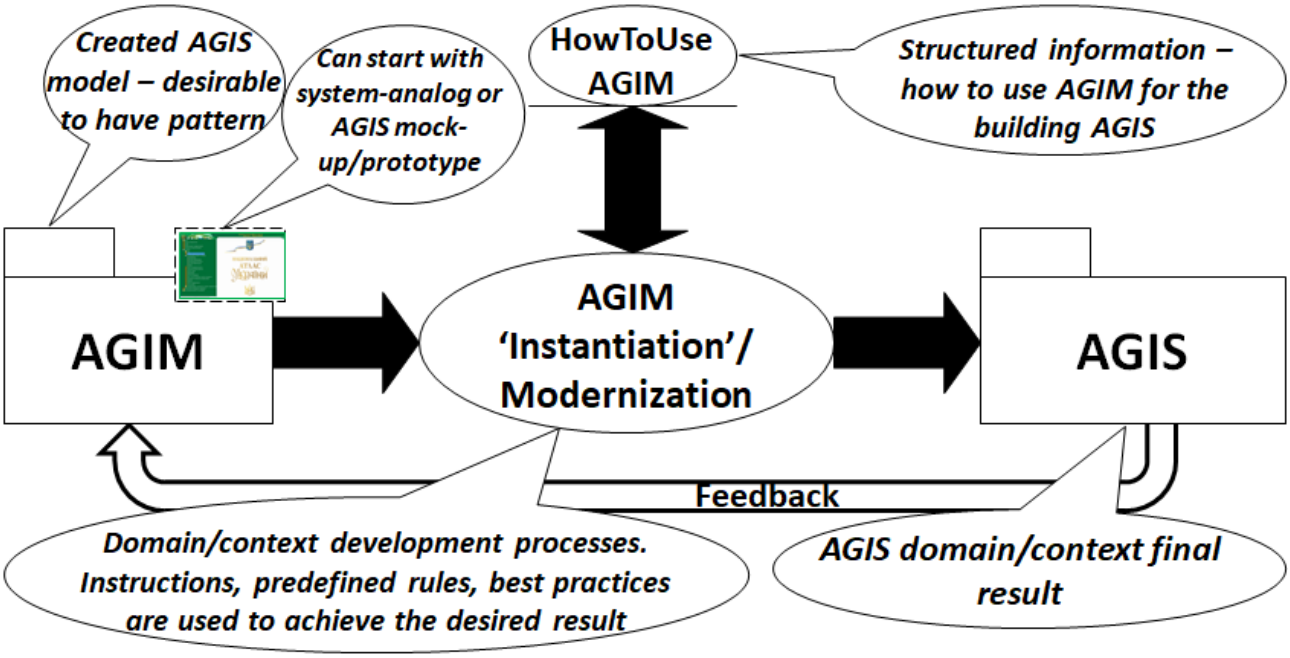

Fig. 6. SoFr usage in the case of activities to create AGIS

Articles [8] describe the product and process conceptual provisions of the AGIS of the Cultural Heritage of Ukraine first queue – AGIS-CH1, compactly shown in Fig. 7.

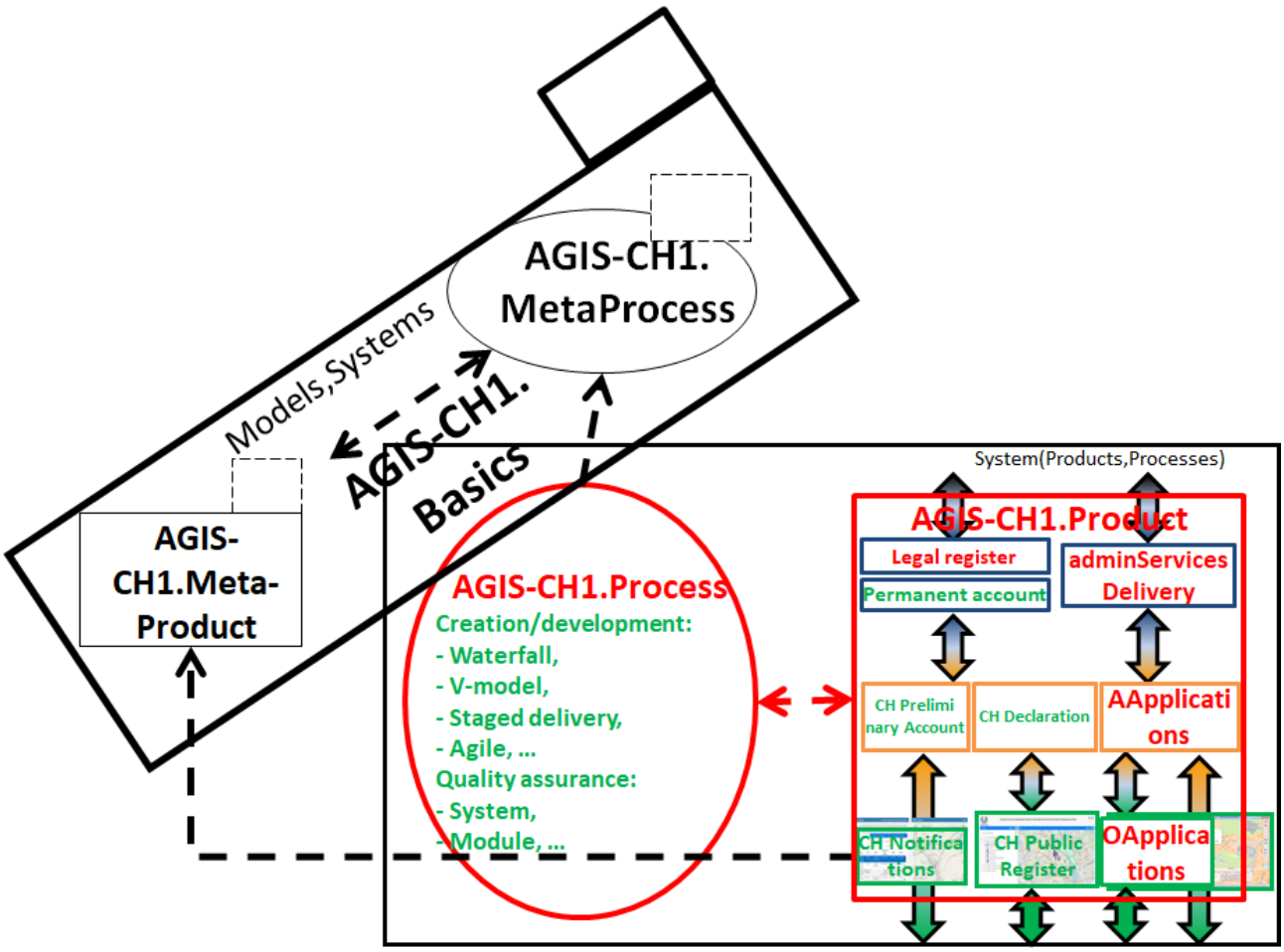

Fig. 7. SoFr usage in the AGIS-CH1 project

AGIS is an acronym for Atlas GeoInformation Systems. This is a very broad class of SpIS, which includes all Atlas Systems implemented by us, and which are practical examples. The term “Practice” denotes an entity that in the “epistemological” hierarchy (hierarchy of knowledge) is in the same echelon as the entities “Method” and “Technology”, as shown in Fig. 4. To interpret the entities, a special class of so-called Large Territory AGIS (LT AGIS) used as an example, where LT can be a country or a “managed” region. AGIS is a hierarchical integrated system of SpIS, where at the lowest stratum of the hierarchy, as a rule, there are Electronic Atlases (EA), and higher-stratum SpIS are certain generalizations of EA, among which are Atlas Information Systems (AtIS), GIS, or GeoInformation Platforms (GIP). The term “certain” here means that the generalization is carried out using such well-known in computer science relations as “classification” or “conformity”. For less formalized relations, the term “metasystemism” is used.

The interest to the methodology of the “Framework Approach to Research and/or Design of Spatial Information Systems” is explained by the need to meet the needs of the so-called “Model-Based Spatial Engineering” (MBSpE) methodology we are creating. It will include three methodics of SpIS extension: Atlas Extension (AtEx), GeoInformation Extension (GIE), Combined Extension, therefore it will de facto also include an “extension methodology”. Let us note again that the “extension” performed for SpIS using the AtlasSF Atlas Solutions Framework. The use of a specific version of AtlasSF is not mandatory, but the use of some SoFr is necessary.

Our methodology will have to operate both as SpIS and also as SpIS in broader sense (SpISb). A possible BMSpE methodology will be part of Model-Based Systems Engineering (MBSysE) [9], which will take into account Model-Based Software Engineering (MBSE) in the statement [10] (we cite both references due to some grammatical errors in the second edition).

We already have experience in using the “practice” of creating a hierarchical (complex) SpIS, so we can quite clearly describe the process of creating this hierarchical SpIS, which both definitely exist. And, if there is both a process and a practice, then there must be a methodology. We believe that by the beginning of 2026 it only needs to be further defined and described. This is enough to use the term “approach” with justification and call it (like the strategy) constructive.

At the moment, we can also call our approach “Framework approach to the extension of (Atlas and Geographic Information Systems)”, without stopping to explain the reasons for such a name. It is based on framework methods from two groups, each homogeneous with respect to the subject: 1) Conceptual Frameworks of X, 2) Solutions Frameworks of X[Y]. Although the approach itself cannot be called a set of homogeneous methods, since in the first case the subject is X, and in the second – X[Y]. In general, frameworks are solutions obtained by applying one of the two or both frameworks (and an arbitrary number of times). The concept of X has been explained several times before. The concept of X[Y] is divided into two concepts with variable values. Y shows the dependence on the value of the (hierarchical) stratum, so the XY SoFr can be called as follows: [General (γ) |

Conceptual (β) | Application (α) | Operational (ω)] Solutions Framework of Stratum Y= γ, β, α, ω, of Subject X.

It should be noted that the results of this article show the ambiguity of a possible methodology for dealing with SpIS. At least two viewpoints should be taken into account: the specialization of the Microsoft meta-methodology and the “direct” application of Model-Based Software Engineering, which is described in the monographs [9] and which is discussed below.

Since the beginning of the century, we have used instances of both Frameworks (CoFr and SoFr) in many projects of creation or exploitation of subjects X and/or X[Y]. In fact, immediately after the discovery of the first SoFr, generalizations began, which can be combined into two directions, homogeneous according to some criterion: 1) subject-oriented, or 2) process-oriented.

Research results - MDSE Methodology through the "prism" of the Framework Approach

Model-Based Engineering (MBE) is defined as: “An engineering approach that uses models as an integral part of the technical baseline that includes requirements, analysis, design, implementation, and verification of system and/or product properties throughout the life cycle of its acquisition” [11]. Note that all “serious” system/product life cycle models necessarily include the phases of research, development, and support. MBE states that appropriate system models must exist for each of these phases.

The MBSysE (Model-Based Systems Engineering) Methodology is defined through a process, method and tool (technology) as follows [11]:

- Process – A logical sequence of tasks performed to achieve a specific goal. A process defines “WHAT” must be done without specifying “HOW” each task must be performed.
- Method – Consists of techniques/ways/means of accomplishing a task, the “HOW” of each task. The terms “method”, “techniques/ways/means”, “practice” and “procedure” can be used interchangeably in this context.

- Tool – A tool used in a particular method that can improve the effectiveness of a task. Thus, methods help bridge the gap between process and tools. The purpose of a tool should be to facilitate the execution of the “HOW”.
- Methodology – Defined as a set/collection of related processes, methods, and tools.

The methodology of Model-Based Systems Engineering (MBSysE) is briefly described in [1] following [11]. Here we briefly review the methodology of Model-Driven Software Engineering (MDSE) using the monographs [10]. Obtained from [10; fig. 2.1] Fig. 8a visually represents the relations between, in essence, the approaches to modeling, which in English are denoted by the acronyms M*. Here we are most interested in MDSE, which is a variant of MDE. However, we should not forget about Fig. 8b and Fig. 8c.

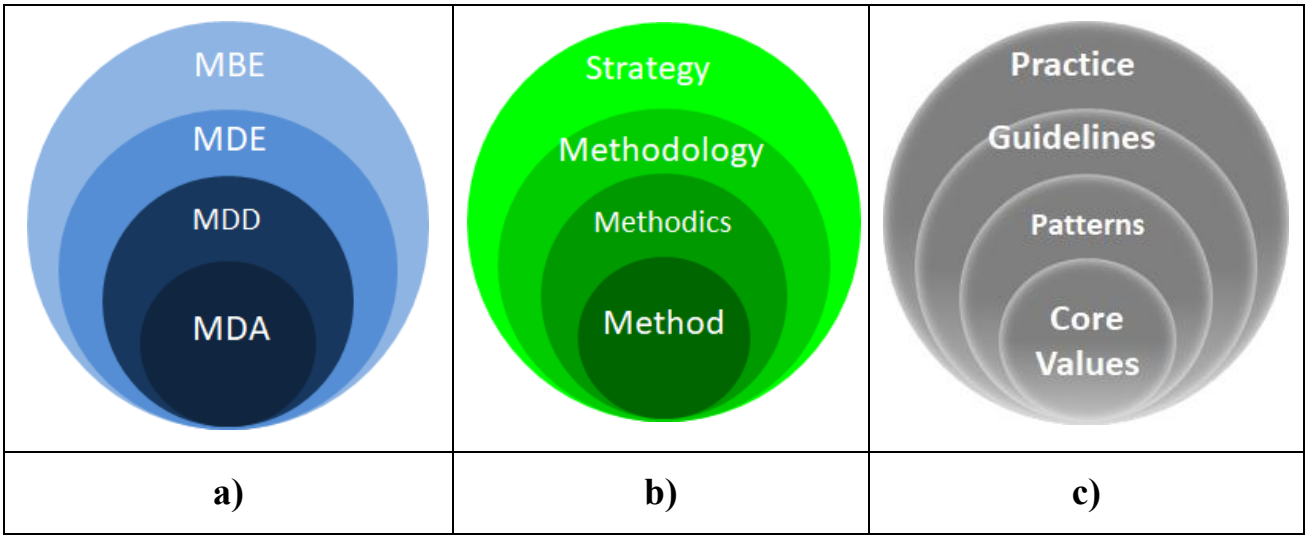

Fig. 8. Hierarchy of notions in different activity spheres: a) Relations between different acronyms M* [10; fig. 2.1]; b) hierarchy of entities of the Framework Approach; c) hierarchy of PBE concepts

Fig. 8b shows the relations between the echelons of the Framework approach. They correspond to the strata of the modeling system: General (γ), Conceptual (β), Application (α), Operational (ω). Fig. 8c shows the relations of the main concepts of the so-called Pattern Based Engineering (PBE). In the original [12] it was called “A model bringing together the key elements that support our PBE efforts”.

Original MDSE

According to [10], MDSE can be defined as a methodology[1] for applying the benefits of modeling to software engineering activities. In general, the methodology includes the following aspects: 1) *Concepts*: The components that form the methodology, ranging from linguistic artifacts to actors, etc.; 2) *Notations*: The way to represent concepts, such as the languages used in the methodology; 3) *Process and rules*: The actions that lead to the production of the final product, the rules for their coordination and control, and the statements about the desired properties (correctness, consistency, etc.) of the products or process; 4) *Tools*: Applications that simplify the execution of actions or their coordination, covering the production process and supporting the developer in the use of notations.

In the context of MDSE, the *main concepts* are: models and transformations (i.e., operations for manipulating models). All these concepts are interconnected, as follows from the famous equation of Niklaus Wirth: *Algorithms + Data Structures = Programs*. In the context of MDSE, the simplest form of this equation would look like this: *Models + Transformations = Software.*

Model-Driven Development (MDD) is a development paradigm that uses models as the primary artifact of the development process. Typically, in MDD, implementations are (semi)automatically generated from models.

Model-Driven Architecture (MDA) is a specific vision of MDD proposed by the Object Management Group (OMG) and thus based on the use of OMG standards. Therefore, MDA can be considered a subset of MDD where the modeling and transformation languages are standardized by OMG.

On the other hand, MDE can be a superset of MDD because, as the E in MDE suggests, MDE goes beyond pure development activities and encompasses model-based tasks of the complete software engineering process (e.g., model-based system evolution or model-driven reverse engineering of a legacy system).

---

[1] We recognize that methodology is a loaded term that can have multiple interpretations. In this context, we use the term not to refer to a formal development process, but rather to a set of tools and guidelines, as defined in the text above.

Finally, the term “model-based engineering” (or “model-based development”) used to refer to a softer version of MDE. That is, the MBE process is a process in which software models play an important role, although they are not necessarily the key artifacts of the development (i.e., they do NOT “drive” the process, as in MDE). An example would be a development process where, in the analysis phase, designers define domain models of the system, but then these models are directly passed on to programmers as blueprints for manual code writing (without automatic code generation and without explicit definition of any platform-specific models). In this process, models still play an important role, but are not central artifacts of the development process and may be less complete (i.e., they can be used more as blueprints or sketches of the system) than those used in the MDD approach. MBE is a superset of MDE. All model-driven processes based on models, but not vice versa.

MDSE provides a comprehensive view of (software) system development. Fig. 9 provides an overview of the main aspects considered in MDSE and summarizes the way in which different issues are addressed. MDSE seeks solutions according to orthogonal dimensions (directions): conceptualization (columns in the figure) and implementation (rows in the figure).

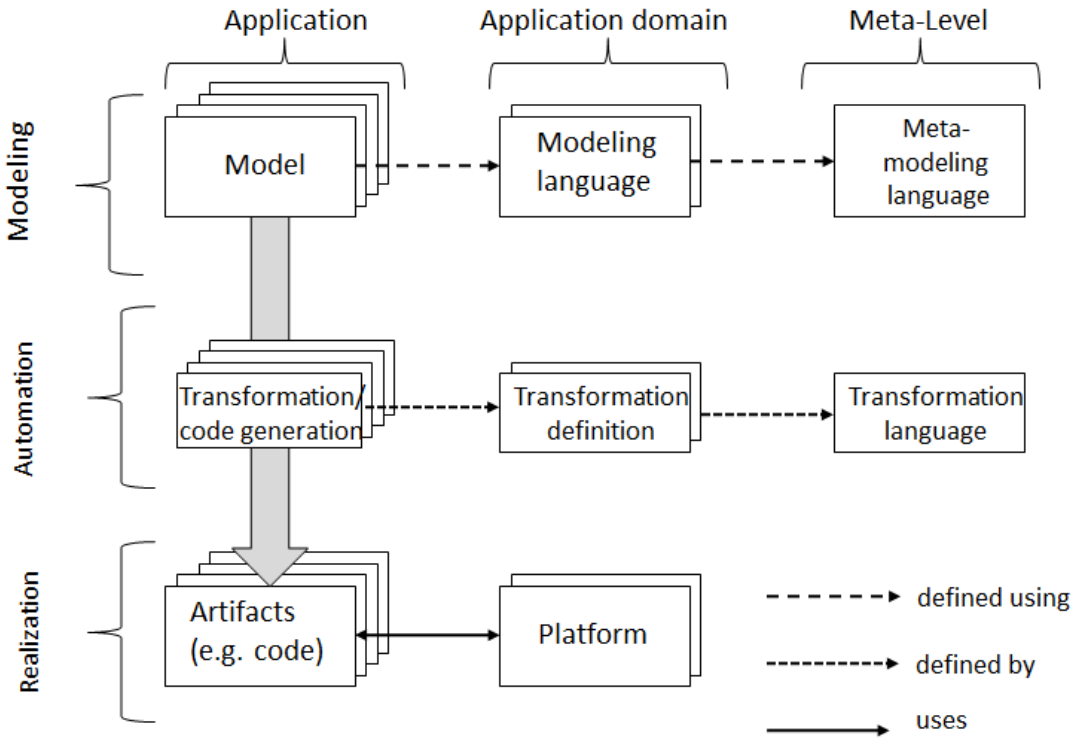

Fig. 9. Overview of the MDSE methodology [10; fig. 2.2]

MDSE and its CoFr

Fig. 9 can be represented using the Conceptual Framework, in this case the CoFr of the MDSE. But here we do not do this so as not to distract unnecessary attention from the figure [10; fig. 2.2]. Instead, we recall [1; Fig. 2], which shows orthogonal relational and subject dimensions. They correspond to the dimensions of conceptualization and implementation from Fig. 9. We also recall that in the Conceptual Framework of Relational Cartography [6] these same dimensions called Epistemological and Transformation, according to which the same relations formed. To the further description of the MDSE from [10] we add what is related to the Conceptual Framework.

Implementation (transformation in CoFr) deals with the transformation of models into some existing or future working systems. Therefore, it consists of defining three main aspects. At the same time, transformation in CoFr can be carried out at any of the four layers: general, conceptual, application and operational.

- Modeling level: where models are defined.
- Implementation level: where solutions are implemented through artifacts that are actually used in working systems (this consists of code in the case of software).
- Automation level: where mappings from modeling to implementation levels are made.

Conceptualization is focused on defining conceptual models for describing reality. It can be applied at least at three main levels (strata). We recommend paying attention to Fig. 2, which shows that CoFr represents the modeling of three worlds: Abstract (virtual), Abstract-physical (digital), and Physical.

- Application layer (Physical world in CoFr): where application models defined, transformation rules executed, and actual working components generated.
- Application domain level (Abstract-Physical World in CoFr): where the modeling language, transformations, and implementation platforms for a specific domain are defined.
- Metalevel (Abstract World in CoFr): where the conceptualization of models and transformations is defined.

The main flow of MDSE is from application models to the current implementation, through successive model transformations. This allows for the reuse of models and the exploitation of systems on different platforms. Indeed, at the implementation level, the running software uses a specific platform defined for a specific application domain.

To make it possible, models specified according to a modeling language, which in turn defined according to a metamodeling language. The execution of transformations is determined based on a set of transformation rules defined using a specific transformation language.

In Fig. 9, system *design* is carried out according to a top-down process from *normative models* that define how the scope is limited and how the goal is to be implemented. On the other hand, abstraction is used bottom-up to create *descriptive models* of systems.

MDSE CoFr and Typical model transformation pattern

Finally, we will give an example of using the Conceptual Framework of the MDSE and the so-called Typical model transformation pattern [13]. The emergence of this subsection is explained by the fact that above in the subsection “Original MDSE” model transformation is called the second of the two main concepts. The Typical model transformation pattern is shown in Fig. 10.

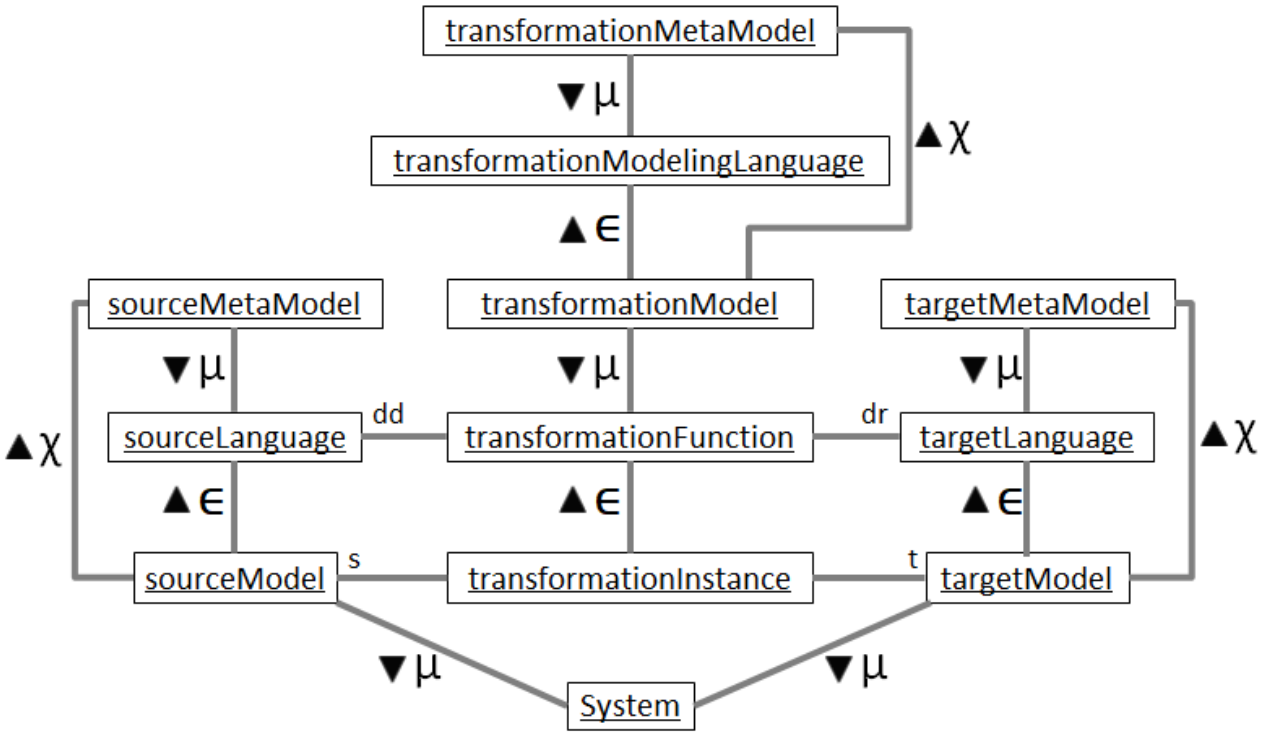

Fig. 10. Typical model transformation pattern [13; Fig. 8]

The issue of model transformation intensively considered in the literature on MDSE. Let us mention only one, but fundamental, work [14]. Although we did not investigate the “typicality” of the given Favre pattern. We only have to say that both the article [13] itself and its subsection “4.2. MDE patterns” turned out to be very useful in our research and their justification. Here it is worth mentioning [13; Fig. 7], called “The "meta-step" pattern and transformation rule”. The meta-step pattern describes one of the fundamental relation, called “conformity” and significantly expands the understanding of the “instantiation” relation. That is, at least the “meta-step” pattern is typical.

The fact is that the “conformity” relation used to the hierarchy of system models, which is what we needed. The “instantiation” relation historically used to the hierarchy of objects. It is, perhaps, together with the inverse classification relation, applicable to systems. However, the instantiation relation “coarser” than the conformity relation in describing the relations between systems. For further explanation, we will need a representation of the hierarchy of strata of computer system patterns, which we often use when creating atlas systems (Fig. 11).

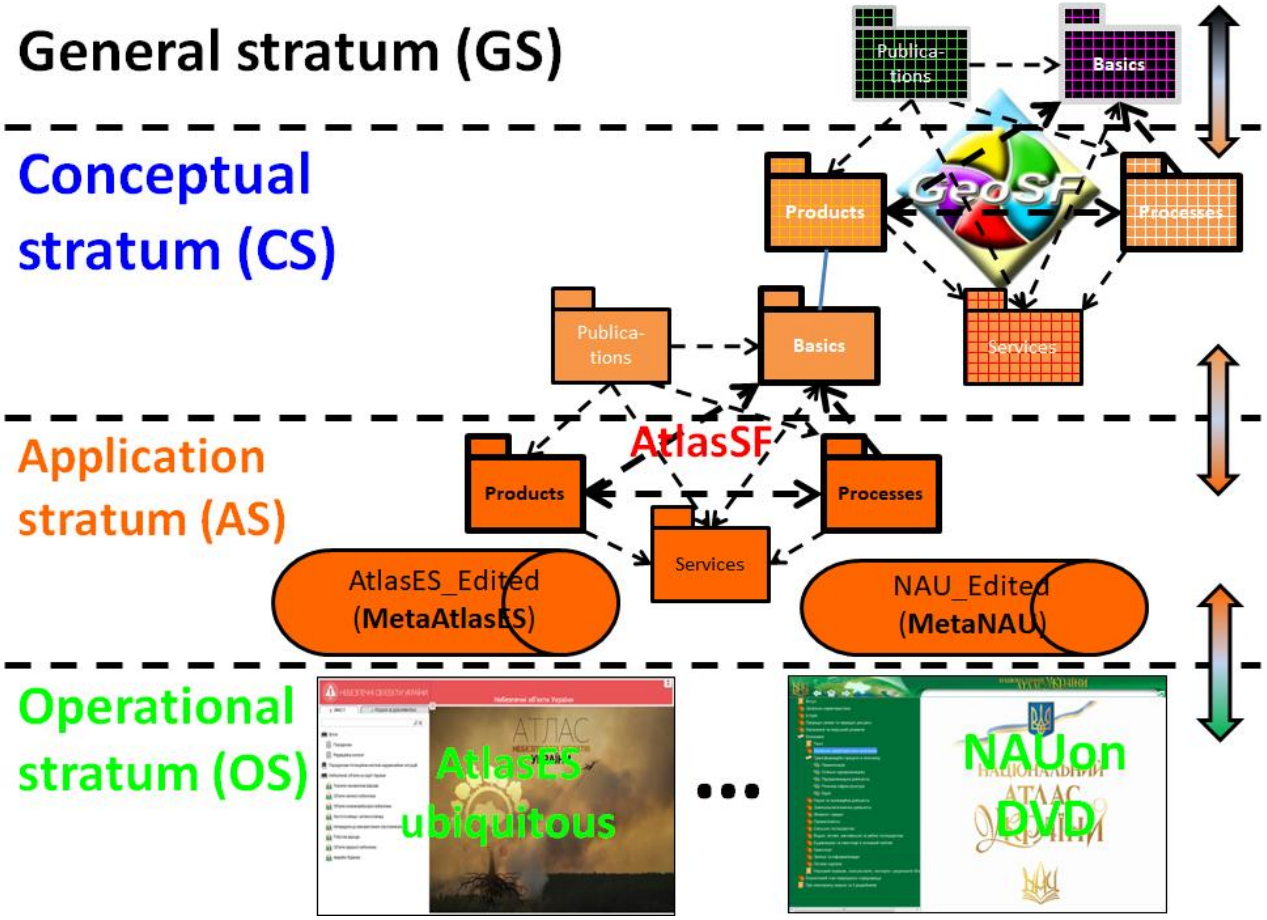

Fig. 11. Representation of CoFr strata and corresponding GeoSF and AtlasSF SoFr

A full description of the transformation pattern usage (Fig. 10) is contained in the monograph [7]. For this article, we note the following (in Fig. 12, the notations used are: tr=transformation, D=DataLogic, I=InfoLogic, U=UsageLogic, α=Application Layer, ω=Operational Layer).

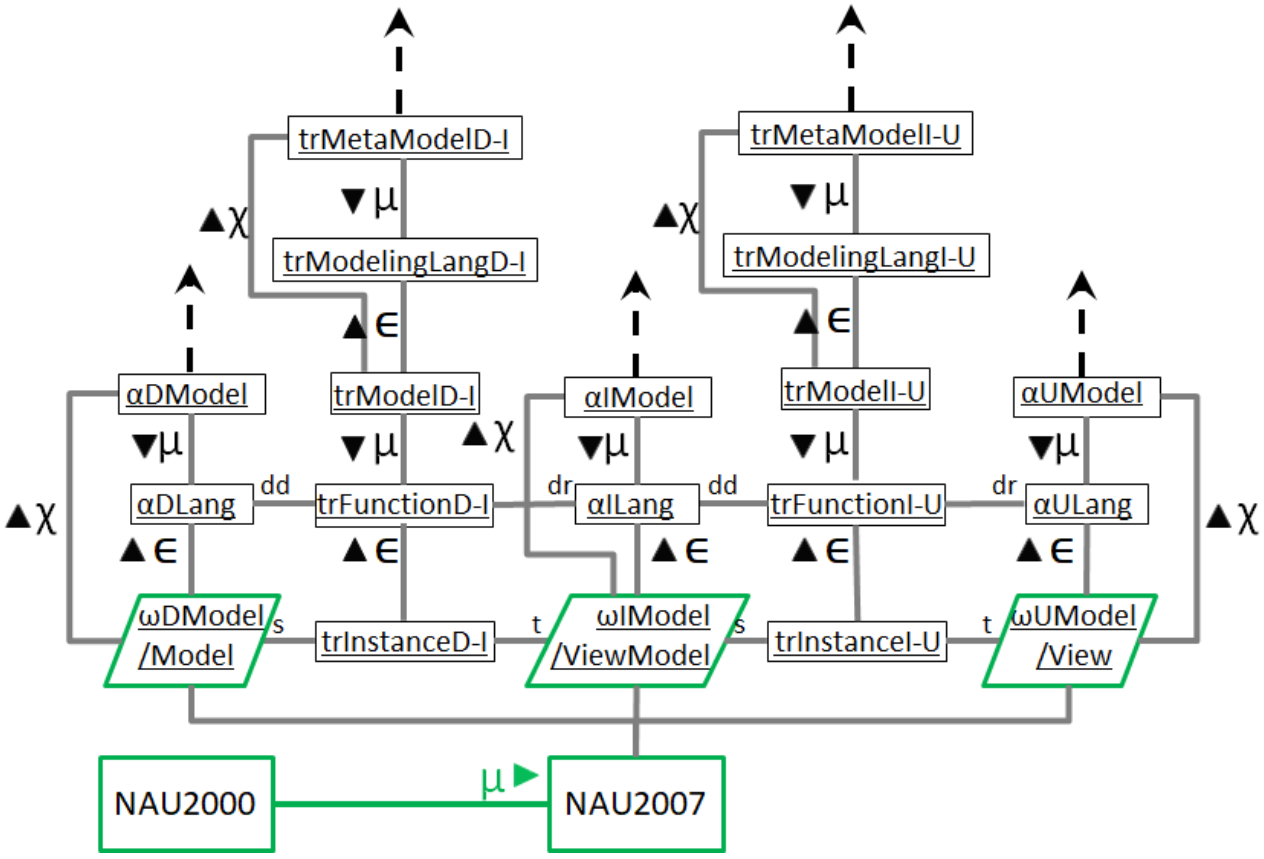

Fig. 12. Typical model transformation pattern in the ElNAU project

At the lowest, Operational, stratum, three models shown, from left to right: Data Model ωDModel, Information Model ωIModel, Usage Model ωVModel). They correspond to the MVVM pattern: Model, View, ViewModel. Here, an analogy used with the process of building a map, according to which, from the data obtained in the first step, in the second step the data representation (image) is designed, which is reduced to setting the scale, explaining the notations used in the image, etc. In the third step, the map provided to the user with the necessary operations performed.

The entry ElNAU2000n μ▶ ElNAU2007n means that the pilot version of ElNAU, called the Atlas of Ukraine 2000 (ElNAU2000 in the narrow sense - ElNAU2000n), was used to model the first version of ElNAU - ElNAU2007n. ElNAU2007n was released in a circulation of 5,000 thousand copies in 2007. The letter "n" used to separate ElNAU in the narrow sense (ElNAUn) from ElNAU in the broad sense (ElNAUb).

Conclusions

The work proves the correspondence between the "neutral" methodology of software development and the methods of the future methodology of the Framework Approach. For this, the method of Conceptual Frameworks used for the subclass of SpIS - software systems, which "include" software. The methodology will correspond to the so-called Model-Driven Software Engineering (MDSE), and the Conceptual Framework is the first of the two systemic methods of the Framework Approach.

References

1. Dyshlyk, O., & Chabaniuk, V. (2025). Karkasnyi pidkhid yak stratehiia doslidzhennia proektuvannia skladnykh prostorovykh informatsiinykh system (na prykladi NIHD) [Framework approach as a research strategy for the design of complex spatial information systems (using the example of NIGD)]. *Land Management, Cadastre and Land Monitoring*, (1), 104–130. https://doi.org/10.31548/zemleustriy2025.01.09

2. Shynkaruk, V. I. (Ed.). (2002). *Filosofskyi entsyklopedychnyi slovnyk* [Philosophical encyclopedic dictionary]. Abrys.

3. Dyshlyk, O., & Chabaniuk, V. (2026). Standartyzatsiia metodolohii MSF Agile z dopomohoiu standarta ISO/IEC 24744 [Standardization of the MSF Agile methodology using ISO/IEC 24744]. *Land Management, Cadastre and Land Monitoring*, (1). (in press)

4. Dyshlyk, O., & Chabaniuk, V. (2026). Zastosuvannia metodyk karkasnoho pidkhodu pry stvorenni prostorovykh informatsiinykh system z vykorystanniam suchasnykh tekhnolohii Microsoft [Application of framework approach methods in creating spatial information systems using modern Microsoft technologies]. *Modern Achievements in Geodetic Science and Production*, 1(51). (in press)

5. Chabaniuk, V., & Dyshlyk, O. (2024). Formalizatsiia kontseptualnoho karkasa prostorovykh system [Formalization of the conceptual framework of spatial systems]. *Land Management, Cadastre and Land Monitoring*, (3), 64–88. https://doi.org/10.31548/zemleustriy2024.03.06

6. Chabaniuk, V. (2024). Karkas atlasnykh rishen AtlasSF yak pidkhid, metod i zasib stvorennia atlasnykh i heoinformatsiinykh system [AtlasSF framework of atlas solutions as an approach, method and means of creating atlas and geoinformation systems]. In *Heohrafichna nauka ta osvita: perspektyvy y innovatsii* (Proceedings of the IV International Scientific-Practical Conference, pp. 197).

7. Chabaniuk, V. (2018). *Relatsiina kartohrafiia: Teoriia ta praktyka* [Relational cartography: Theory and practice]. Instytut heohrafii NAN Ukrainy.

8. Chabaniuk, V., Dyshlyk, O., Polyvach, K., Pioro, V., Kolimasov, I., & Nechyporenko, Yu. (2022). Holovni kontseptualni polozhennia stvorennia elektronnoho derzhavnoho reiestru nerukhomoi kulturnoi spadshchyny Ukrainy. Chastyna 1 [Main conceptual provisions for creating an electronic state register of immovable cultural heritage of Ukraine. Part 1]. *Land Management, Cadastre and Land Monitoring*, (2), 133–154; (3), 114–136. https://doi.org/10.31548/zemleustriy2022.02.11; https://doi.org/10.31548/zemleustriy2022.03.11

9. Holt, J. (2023). *Systems engineering demystified: Apply modern, model-based systems engineering techniques to build complex systems* (2nd ed.). Packt Publishing.

10. Brambilla, M., Cabot, J., & Wimmer, M. (2017). *Model-driven software engineering in practice* (2nd ed.). Morgan & Claypool Publishers. https://doi.org/10.2200/S00441ED1V01Y201208SWE001

11. Martin, J. N. (1996). *Systems engineering guidebook*. CRC Press. https://doi.org/10.1201/9780138737443

12. Ackerman, L., & Gonzalez, C. (2011). *Patterns-based engineering: Successfully delivering solutions via patterns*. Addison-Wesley.

13. Favre, J.-M. (2004). Towards a basic theory to model model driven engineering. In *Proceedings of the 3rd UML Workshop in Software Model Engineering*.

14. Lano, K., & Kolahdouz-Rahimi, S. (2014). Model-transformation design patterns. *IEEE Transactions on Software Engineering, 40*(12), 1224–1259. https://doi.org/10.1109/TSE.2014.2354344

О. П. Дишлик О., В.С. Чабанюк

МЕТОДОЛОГІЯ КЕРОВАНОЇ МОДЕЛЯМИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ У КАРКАСНОМУ ПІДХОДІ

Продовжено дослідження «Каркасного підходу до поводження з (ієрархічними) просторовими інформаційними системами (ПрІС)», далі – Каркасного підходу. До цього розглянуто сутності Каркасного підходу, що відносяться до верхніх ешелонів його епістемологічної ієрархії. А саме, до (назва відповідної ешелону статті наведена в лапках «…»): 1) першого ешелону – «Каркасний підхід як стратегія дослідження і проектування складних ПрІС (на прикладі НІГД)»; 2) знову першого ешелону – «Каркас Рішень Microsoft (КаРі М - MSF) як Узагальнена методологія Каркасного підходу поводження з ПрІС»; 3) другого ешелону – «Стандартизація методології MSF Agile з допомогою стандарта ISO/IEC 24744»; 4) третього ешелону – «Застосування методик Каркасного підходу при створенні ПрІС з використанням сучасних технологій Microsoft». Ешелони об'єднують групи користувачів моделюючої системи. Їм відповідають страти цієї самої системи.

Три з чотирьох згаданих статей суттєво використовують спеціальні знання про MSF і про сучасні інформаційні технології (ІТ) Microsoft. Щоб зменшити залежність від цих знань, у даній статті ми скористалися «нейтральними» знаннями про методології розроблення програмного забезпечення. Для цього через «каркасну призму» розглядається так звана Базована на Моделях Програмна Інженерія (БМПІ). Остання є Базованою на Моделях Системною Інженерією (БМСІ) і, більш загально, Базованою на Моделях Інженерією (БМІ). Їх розуміння потрібне для створення власної методології, яка називається методологією Базованої на Патернах Просторової Інженерії (БППрІ). БППрІ включатиме методики розширення

ПрІС, про які згадувалося у наших роботах раніше. Більше того, вказані методики вже використовувалися нами на практиці інколи без асоціації з якоюсь методологією. При цьому потрібно зауважити, що наші методики розширення ПрІС застосовуються до конструкцій, які вже структуровані в результаті використання так званого Каркасу атласних рішень AtlasSF.

Ключові слова**:** *методологія Керованої Моделями Програмної Інженерії (КМПІ), Концептуальний Каркас КМПІ, методика розширення ПрІС*