

The Fundamentals Regarding the Usage of the Concept of Interface for the Modeling of the Software Artefacts

Dorin Bocu

Transylvania University of Brasov, Romania
d.bocu@unitbv.ro

Razvan Bocu

Transylvania University of Brasov, Romania
razvan.bocu@unitbv.ro

Abstract

This paper presents the conceptual foundations of a software system's solution modelling activity, which is formally based on two essential concepts: the artefact and the interface. This modelling activity envisions two objectives: the explicit emphasis on the interfaces' importance in the software engineering, and the preparation of the framework inside which the loop structure-behaviour can be formalized considering the inherent benefits for the modelling activity in general, and for the modelling activity automation in particular.

Keywords: Artefact, Interface.

1. Introduction

The efficient collaboration of the systems or between the components of a system may be considered a necessary and sufficient condition for the systems to work, in general, while their components keep, partially and temporary, the freedom to look for new modalities to express their internal dynamics. The good understanding of the collaboration's importance for the systemic and systematic explanation and modelling of the reality is essentially connected to the concept of interface. Considering a very broad perspective, we may assert that the interface is a theoretical tool that is needed by any thinker that ventures in the turbid waters of the unknown.

In the context of the software systems, we may consider that the interfaces are not just products that pertain to the knowledge and / or modelling effort, but they are components of the real systems, which are indispensable for their survival and modernization.

It usually happens that considering the relationship between a paradigm and its user, the interface as a concept is not the real challenge, but the ability to identify and specify particular and elegant interfaces, which conserve and partially redefine the equilibrium inside a system.

The interfaces of the real systems are studied and understood considering a certain granularity level, which is specific to the momentary interest, but also to the technologies and the paradigms that are used. In other words, the approximation is also inherent in an explanation or modelling approach that is centred towards interfaces. The error that is characteristic to the process of approximation ultimately determines the practical utility of the explanation or modelling approach that is centred towards interfaces.

In general, the errors that become apparent in a systemic demarche are of two kinds: methodologically assumed and involuntary. The errors that are methodologically assumed represent the necessary ingredients of any iterative systemic approach, which involves the progressive deciphering of the system's complexity considering multiple perspectives that correspond to particular abstraction schemes.

The involuntary errors are provoked by the unsatisfactory preparedness of the modelling specialist, provided that there are no flaws of the methodological context. The preparedness is normally the expression of a solid theoretical background, together with a notable practical experience. As an example, the specialists that do not commit individual errors of any kind are too few to satisfy the requirements of the IT industry. Thus, it can be asserted that the evolution of the IT projects is questioned, even if considering only a theoretical perspective. What should one expect

considering a practical perspective? Naturally, one should expect that mitigation efforts were conducted through a careful management of the IT projects.

This is a proof that the quality of the artefacts that are produced by the IT projects depends on the quality of the engineering activities, but it also depends on the feasibility and reliability of the IT company's management style. These two fundamental types of activities overlap according to some general requirements that are specified in Figure 1, but also in (Bocu & Bocu, 2013). The problem of the process that is described in Figure 1 is simple: how can we limit the pressure of the involuntary errors on the management of the IT project and, obviously, how can we optimize the abstraction method in order to obtain better quality artefacts? The possible answer to these two questions is represented by the modelling that highlights the importance of the interfaces for the software systems engineering. This is the problematic that is thoroughly approached in the following sections.

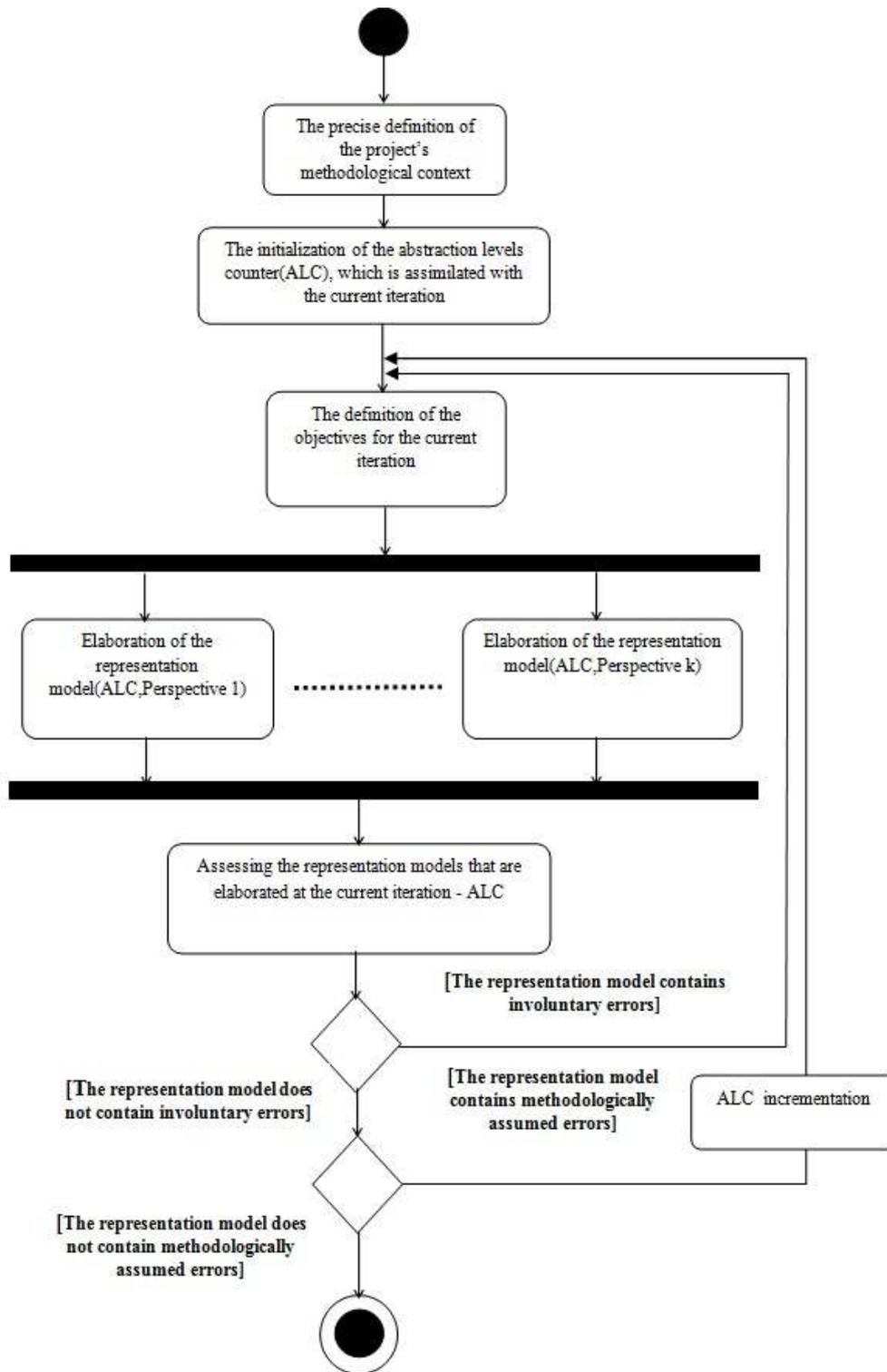
2. The conceptual apparatus that is used in the interface oriented modeling

2.1. Interface Orientation. Introduction

The experience that is accumulated regarding the modelling paradigms in the software engineering is impressive. Thus, the software engineering recognizes modelling paradigms like object orientation, aspect orientation, component orientation, service orientation, agent orientation. In one form or another, these paradigms prove their excellence in certain types of IT projects. At the same time, these paradigms reveal their objective limits when they are used to engineer the real world software systems.

Every modelling paradigm represents, in fact, a modality to represent the real world using a specific formal framework.

The specificity of the formal framework is defined from both a syntactic and semantic perspective. The formal syntactic framework of a paradigm refers to the concepts that are used by the paradigm in order to represent the real world, but also to the recommended principles that allow for these concepts to interact in a correct and efficient manner. Both the concepts and the principles benefit from a formal representation that ultimately favours communication as a secondary modelling lever inside the IT projects. Every syntactic artefact of a paradigm can be associated with a certain real world semantics, which it abstracts. As a consequence, considering that the real world continuously enhances its semantic potential, the syntactic constructs that are favoured by the paradigm may become problematic.



The formal purity of the syntactic constructs that are related to a paradigm, thus becomes an obstacle in the way of the modelling process.

Figure 1. Visual and synthetic representation of the modelling process in the software industry

Even considering that a paradigm possesses the extension mechanisms of its own formal potential, the semantic richness of the real world cannot be always captured without compromises. From here, there is only one step for the paradigm's criticism to appear, which is represented by the methodic wish to use the paradigm without risky formal compromises. Therefore, the logic of the relation between the syntax of a paradigm and the semantics to which

it relates does not exclude the possibility of some criticism to occur, which is based on the coherent assessment of the produced artefacts' quality. Consequently, the attempt to obtain a correct and long lasting definition of interface orientation announces itself as being fundamentally risky. This would not be a novelty since the world of the IT systems is based on both excellently articulated theories, and also on practical development recipes, which value the attractiveness of the intuition and / or the vigour of the empirical approach. This is a valid principle, as each formal system should have its roots in a rich ideatic universe, whose appearance is not necessarily deducible from a limited number of concepts and axioms.

It is certain that the software engineering already uses the concept of interface in order to found other paradigms and useful practices regarding the modelling, implementation and testing of the software systems. The subtlety of the concept of interface has been perceived and appreciated by numerous specialists. This paper describes a modelling method of the software systems that considers the interface as its fundamental concept. The paper (Bocu & Bocu, 2013) presents the concept of interface as an ingredient that plays an important role in the evolution of real and artificial systems, and also highlights the tight connection that exists between the abstraction methods in general and the concept of interface.

The authors of this paper have the intention, which is deducible from Figure 1, to use the interface both as a mediator among the structure and the behaviour of a software system, but also as a maximum utility ingredient in the effort to approach the management towards the process of a software system's solution engineering.

2.2. The Concept of Artefact

2.2.1. Preliminary Remarks

The world that we attempt to understand and to model is made up of objects, which will simply be named artefacts, as it is useful to avoid the interferences with other explanatory or modelling paradigms. The artefact is a material or spiritual object, which has the feature to be unique in a world that highlights its usefulness through the methodic experimentation of the interaction that is centred towards fostering the diversity. It is envisioned to place a greater importance on formalization, so the following definition is proposed.

Definition 1. Let us call artefact an object that is created by a human mind, whose informational features can be captured and represented by using three types of patterns: the architectural pattern, the structural pattern, and the behavioural pattern.

The architectural pattern of an artefact optimally captures and represents data that suggest the usefulness and value of the artefact. It is intended to further insist on the high abstraction level of the architectural patterns, and it is legitimate to say that they reflect the demiurgic intentions of the artefact's creator. The main utility of these intentions is centred towards the idea to improve the environment in which the artefact will operate.

The structural pattern of an artefact optimally captures and represents data that refers the components of the artefact and the relations among them. The environment in which the artefact will operate has to be structurally stable in order to make the most out of the subordinated artefacts. Consequently, it is possible to assert that the structural pattern reflects the compromise between the change offer that is encapsulated in the architectural pattern, and the need for stability of the environment in which the artefact will operate.

The behavioural pattern of an artefact optimally captures and represents data that refers to the dynamics of the relations that express between the components of the artefact. Every behavioural pattern becomes a concrete modality that values the potential that is encapsulated in the structural pattern. It is relevant to mention that an artefact appears during the modelling process in two hypostases: the descriptive model hypostasis, and the instance hypostasis. The descriptive model hypostasis is presented in Definition 1. The instance hypostasis is introduced in Definition 2.

Definition 2. Let us define an instance hypostasis artefact as an artefact that is obtained through the customization of the data that determine the three patterns, which describe any artefact.

The instance hypostasis of an artefact is useful in order to streamline the communication among the members of an IT project. The descriptive model hypostasis remains fundamental in order to specify, represent and manage the attributes of an artefact.

2.2.2. The Concept of Interface

The general theory of the systems has been created in order to explain and model the real system, which interact with the environment that contains them. The interaction between systems and the environment that includes them is accomplished by resorting to the use of interfaces. Considering this perspective, any artefact that operates as a substitute for a certain system possesses one or more interfaces that allow it to interact with other artefacts.

Definition 3. Let us call interface of an artefact a component of the artefact that abstracts part of its interaction potential with the environment into which it integrates.

The artefacts, as components of a problem's solution in the software industry, collaborate in order to ensure the efficient operation of the integrative system. The diversity of the types of artefacts that are developed with the goal to elaborate and represent the solution of a problem in the software systems' engineering contributes to the diversification of the hypostases that determine the operation of the concept of interface. Without considering this diversity, certain traits of the interfaces are essential in order to structure around them a particular approach. It is all about the following:

- The imperative stability, which is methodically assumed, is beneficial for the engineering process, but also for the management process;
- Fostering the specification and realization of some artefacts that are optimally coupled with each other;
- The optimal configuration, as a modality to highlight the artefact's structural and behavioural potential;
- The openness towards testing, regardless of the type of interface.

Although it is extremely useful and important as a pure abstraction, an interface also showcases its operational usefulness if we consider another two common aspects that pertain to the work with interfaces: the implementation and the valorisation.

Definition 4. Let us call implementation of an interface a circumstantial modality to provide the necessary support in order to instantiate the interface.

As soon as the problem of an interface implementation is addressed, it makes sense to talk about the valorisation of one of its instances.

Definition 5. Let us call valorisation of an interface a pragmatic usage scenario of one instance of the interface.

The separation of the interface from its circumstantial implementations brings flexibility for the process of a software system's realization, but also regarding the endeavour to increase the qualitative level of the software system. Furthermore, the separation of the interface from its potential valorisations is an advantage, which is appreciated by any IT specialist that is preoccupied by the impact of any changes that may influence the software systems' lifecycle.

2.2.3. Further Remarks Regarding the Concept of Architectural Pattern of an Artefact

In practice, the architectural pattern represents the modality that is chosen by the artefact's creator in order to answer the question "What does the artefact do?". As an example, let us suppose that the artefact is a new educational system, so it is obvious that we cannot circumvent the question "What does the new educational system do?", without taking the risk on multiple plans. Although it may seem peculiar and incommodious to those that do not believe in the practical

usefulness of the abstract ideas, the answer to this question is never simple. In fact, the answer to the question "What does the artefact do?" reflects the ability of its creator to imagine the essential properties of the artefact. Let us consider the example of a new educational software system. Thus, it may be considered that its essential properties are deducible from the answers to the following two questions:

- What is the mission of the educational system (that is to say, what will be its contribution regarding the transformation of the beneficiary society?);
- What is the vision that is considered by the creator during the endeavour to realize the system?

Although it is a truism, it is significant to remark that the definition of an artefact's mission and the vision that is promoted on the occasion of its realization bears the seal of subjectivism and the limits that stem out of this subjectivism.

The biggest problems of an artefact have their source in the modality that the creator used in order to associate the artefact with a mission and with a vision. The artefact is not appreciated by the society according to its technical complexity, but based on its perceived influence on the society, and this influence is advisable to be beneficial. Therefore, the mission of an artefact and the vision that is associated to the process of the artefact's realization have to be defined correctly and in an inspired manner. If we admit that it is of interest to answer the question "What is the mission of the educational system?" then we assume a very complex endeavour. The complexity is the one that invites us to be prudent, to choose a proper method, and to critically examine considering, at least, three levels: ethics, aesthetics, and pragmatic. The unconfessed goal of any endeavour of this type is represented by the hypothetical absolute truth.

Is the human being able to operate with the absolute truth? This is to say, the truth that is not dependent on the logical infrastructure that is used to discover it.

Is it possible to materialize the absolute truth or is it just an imperfect guiding principle, which guarantees the predictable behaviour of the systems that value the importance of the freewill? It is not applicable to insist in this paper, but it is obvious that without the freewill (the initiative) being able to express, the systems are condemned to stagnate. It seems that there are few creators that are motivated by the idea to live in a world that is methodically preoccupied by the absolute stagnation.

2.2.4. Further Remarks Regarding the Concept of Structural Pattern of an Artefact

While the architectural pattern of an artefact emphasizes the role of the major abstractions in order to understand the artefact's utility, the structural pattern insists on the detailed description of the artefact's potentialities. Considering a structural perspective, an artefact is a potentiality that waits to be valorised directly by the behavioural pattern, and indirectly by the artefact's user. There are a few essential perspectives that should be considered during the abstractization effort that is necessary in order to elaborate the structural pattern:

- The proper scaling of the artefact's resources requirements (what are the necessary and sufficient resources to ensure the accomplishment of the artefact's mission?);
- The systematic minimization of the redundancies that may affect the quality of a design effort, in general;
- The provision of an adequate support in order to make the structure of the artefact compatible with the collaboration intentions of other artefacts;
- The careful modularization of the structural pattern, and consequently of the artefact.

The decisions that are taken during the elaboration of an artefact's structural pattern are prone to an earlier obsolescence than the architectural pattern that is properly elaborated. Thus, the costs to maintain the structural pattern are higher. In theory, every valuable artefact produces a positive change in the environment inside which it operates. The accumulation of these changes environment-wide is the main factor that determines the obsolescence of an artefact's structure.

The structure needs to stagnate in order for it to behave plenary. The effects that are produced by an artefact on the environment announce new changes, even considering the vision that is associated to that artefact.

The essential resources that are used to elaborate the artefact's structural behaviour are:

- The interfaces of the artefact;
- The interfaces of the artefact's components;
- The components of the artefact, considering various levels of specification and abstractization;
- The relations that exist among the artefact's components;
- The relations that exist among the artefact's interfaces;
- The relations that exist between the artefact's components and the interfaces.

It is probably obvious the relativism that pertains to the usage of the words artefact and component. In fact, the distinction is dictated by formal reasons, and not by semantic requirements. Considering a closed semantic universe, the artefact is not always assimilable to a component, while a component is always assimilable to an artefact. Naturally, considering most of the cases, the universe of the artefact is broader than the universe of the component in the context of the communication's logic.

2.2.5. Further Remarks Regarding the Concept of Behavioural Pattern of an Artefact

The basic components of the environment inside which the artefact operates are: time, space and the other artefacts and natural systems. Every artefact, considering its goal and the vision that is associated to it, will apply a certain behavioural pattern, considering the idea that it has to optimize or preserve certain environmental equilibria. In theory, at least, things should be different. All the ideas or important truths to which humans relate target the enhancement of certain systems in which the humans live or act. Naturally, the very concept of enhancement is debatable but, it is not within this paper's scope to discuss on this matter.

Thus, let us define the behavioural pattern of an artefact as a compact and intelligent representation of the artefact's potential of expression.

The behavioural pattern has to efficiently and intelligently harness this potential. The practice shows, through numerous examples, that the human mind imagined a lot of artefacts that haven't come to prominence because their potential hasn't been properly harnessed. This means that there are certain constraints that accompany the approach to harness the artefact's structural potential. The following realities are part of the short list that determines the already mentioned constraints.

- The existence of a supporting technological context in order to effectively realize the artefact. The architecture and the structure may be theoretically approached without considering the technological context. As soon as the behaviour is considered, it is useful to assess the technological context in order to make sure that the artefact's realization becomes reality with reasonable costs. The specification of the artefact's architecture without considering the technological context implies that the effort to realize this technological context is assumed. This implies the existence of significant production costs, which only projects with satisfactory funding can afford.
- If there are several technologies available, the choice of the optimal technology is an important factor.
- The quality of the human resource that is implied in the effective realization of the artefact.
- The management style that is enforced during the artefact's realization.

In fact, it is correct to assert that these constraints are present during the whole process of the artefact's realization. If these constraints are not properly managed, the artefact will be characterized by a questionable quality. The simple problem is: What is the correct approach when trying to direct the transition from the structural pattern that features intentions of stability to the behavioural pattern, which is forced to keep up with the technological evolution, but also with the changing requirements of the artefact's end users? This is the most important challenge that all the creators of

artefacts face: How do their artefacts manage the process of change? In this respect, the human artefacts are of two types:

- naturally resistant to changes;
- naturally dependent on changes.

The artefacts that are naturally resistant to changes are generally methods to understand and model the universe, whose theoretical foundations are characterized by a remarkable stability. This category includes the vast majority of the methods that are used to obtain knowledge in the natural sciences (mathematics, physics, chemistry, etc.). As much as the knowledge progresses extensively and intensively, the scientific methods that are specific to the natural sciences may be adjusted. Nevertheless, these adjustments rather pertain to the precision of knowledge than on its order of magnitude. As an example, which is well known in physics, the special theory of relativity followed by the general theory of relativity have not eliminated the truths that pertain to the Newtonian mechanics, but rather helped at explaining the universe considering a totally different scale. Furthermore, the same assertion may be made regarding the relation that exists between the Euclidean geometry, and the so called Non-Euclidean geometries. The artefacts that are naturally dependent on changes are tightly connected to the technologies that are developed in order to deepen and validate the methods that generate knowledge, which are naturally resistant to changes. Naturally, we simply observe that the computer science is largely based on artefacts that are naturally dependent on changes. The theoretical foundations of the computer science are closely related to the technological foundations. Considering that in the world of technologies the change is the trick, which is used by the industry in order to perform on the long term, the IT artefacts continuously address the problems that are generated by the fragility of the technological solutions. Thus, we are able to conclude, once more, that the value of an IT artefact has to be assessed considering two perspectives: the process that generates the artefact, and the product that is obtained as a consequence of this process. The key to getting the momentum in the fight with the IT technologies changes resides in the re- search of the approach that is associated with the process that produces the IT artefacts.

2.3. Phases Regarding the Realization of an IT Artefact

2.3.1. The Artefact as a Metaphor

In general, a metaphor resembles to a seed that arrives on a fertile land and grows into a tree, whose branches delight the eyes and the curiosity. Regardless of its talent, an artist cannot stay famous to the posterity if the metaphor is not cultivated. The metaphor is a kind of armour in which the artist dresses an idea in order to enhance its beauty, meaning and resistance to degradation that invariably threatens any idea, which witnesses the lapse of time. The scientist pragmatically pursues details and formalisms that precisely describe how they link together, and they have consequently largely forgotten about the importance of metaphors. Ultimately, the metaphor feels at ease at the border between the light and the darkness, the knowledge and the unknown, the moment and the infinity. These hypostases allow the average man or the scientist to metaphorically formulate the relation that exists between the certainties and the uncertainties, which accompany their cognitive approaches.

An artefact that exists as a metaphor represents an attractive, realistic and synthetic statement regarding an idea that aims to improve an existing system.

In other words, the artefact that is just a metaphor has to be interesting for its potential beneficiaries, and it also has to be appealing to the ones that will make this metaphor germinate. It can be observed in Figure 2 that if the artefact is just a metaphor, it can be represented, at least, through a codename and a story.

2.3.2. The Artefact as a Black Box

The accumulation of energy that exists in each ingenious metaphor is progressively released, thus contributing to the transformation of a theoretical promise into effective reality. The artefact successively goes through several maturation stages, as the creator is preoccupied with obtaining an as precise and as close as possible description of the artefact as it exists as a metaphor. The completion of these successive stages is achieved through a methodic abstraction process, while leaving open the possibility to innovate and targeting three main objectives:

- broadening the abstraction scope;
- adding new details;
- detecting and eliminating abstraction errors.

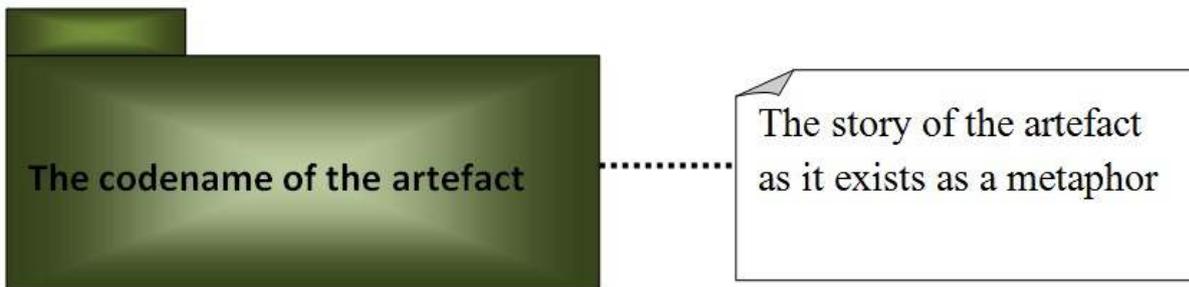


Figure 2. The UML representation of the artefact as it exists as a metaphor

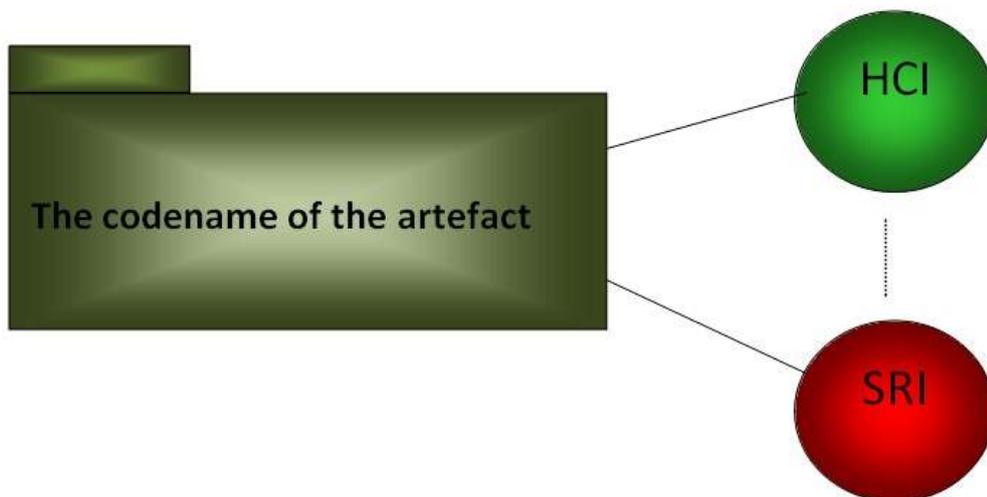


Figure 3. The artefact as it exists as a black box

Consequently, the artefact as it exists as a black box can be represented according to the representation in Figure 3.

It can be noticed that the artefact as it exists as a black box begins to interact with the environment. Two main categories of interfaces may be utilized by any artefact in order to interact with the environment:

- Human Computer Interfaces (HCI);
- Shared Resource Interfaces (SRI).

Two remarks are necessary. They are redundant for specialists, but still necessary for those that begin their journey as creators of IT artefacts. If the artefact's complexity requires, several HCI interfaces may exist, that are eventually structured considering the communication demands of the artefacts with the environment. The assertion is analogous for the SRI interfaces. At this stage, the approach to specify the two categories of interfaces is mostly oriented towards the semantic

aspects of the interfaces. These are deducible following a rigorous analysis of the artefact's potential users. The eventual commitments that are assumed for the artefact as it exists as a metaphor have to be considered when these interfaces are specified. It should be paid attention, even considering the specification variant that is mostly semantic, that these interfaces help to collect certain vital data for the design of the artefact's behavioural pattern. This is accomplished starting with the information that is gathered considering the two initial patterns: the architectural pattern and the structural pattern.

2.3.3. The Initial Structuring Stage

The way the humans understand the world's complexity reflects on their modality to act as creators. Thus, considering that the world's complexity is progressively perceived, so is the artefacts' complexity progressively assimilated and coded. As soon as the artefact's creator specified the intentions regarding the problem to be solved (the metaphor stage and the black box stage), he has to start the endeavour to address this problem. In practice, along with the incipient structuring phase, the creator begins to act as a demiurge. During this phase, the artefact's creator looks for an optimal structuring modality, considering a high abstraction level, such that:

- The interaction of the artefact with the host environment respects the promises of the metaphor;
- The artefact possesses a proper quality behaviour as long as the artefact's structure does not change;
- The artefact adapts, with reasonable costs, to new requirements. It is ideal for the adaptation to new requirements not to affect the structural foundations, especially those that pertain to the initial structuring stage.

The experience teaches us a lot regarding the initial structuring stage of an artefact. Essentially, this is about choosing the optimal architecture, but specialists know that this choice represents a story that a knowledgeable person could hardly approach in a voluminous book. Thus, this paper presents the essential landmarks concerning the design of an artefact during its initial structuring stage. Figure 4 clearly demonstrates that the artefact as it exists as a black box has to be structured considering a high level of abstraction, while insisting on modularization, with all the benefits that are implied by such an approach.

The attention that is paid to the interfaces is not symbolic, as it could be inferred from Figure 4, but it rather involves starting the difficult process that allows for the selection of the interfaces that represent the optimal compromise between the following three types of requirements:

- Optimal user comfort;
- Optimal comfort concerning the change of the implementational support;
- Reasonable costs.

Although it would sound nice to ask the interfaces to offer comfort when they are partially redefined, we would rather not problematize this aspect at this moment.

2.3.4. The Advanced Structuring Stage

The advanced structuring stage is the expression of the thorough mastery of an artefact, and it describes to the last necessary detail the artefact's infrastructure, using adequate concepts, with an explanatory and operational value. The advanced structuring stage is progressively obtained, considering at least two reasons:

- The quantitative and qualitative complexity of the artefact;
- The extensively experimented and studied human habit to gradually approach the unknown.

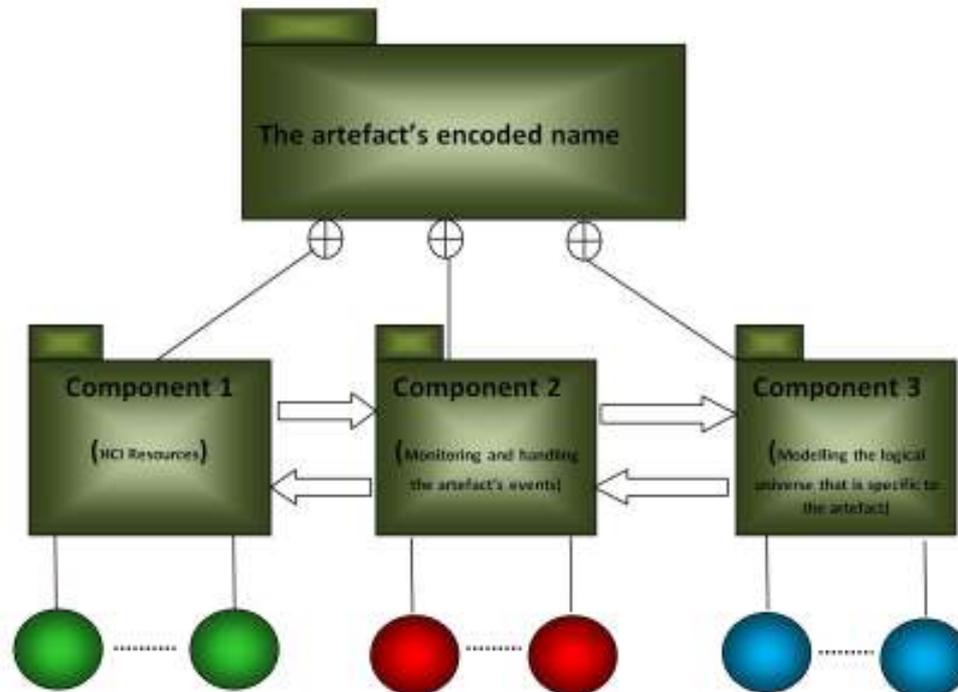


Figure 4. The artefact during the initial structuring stage

The IT developer elaborates the detailed structure of the artefact, while considering several aspects:

- The resources that are necessary to the artefact in order to accomplish its mission;
- The artefact's resistance to the changes regarding the functional requirements;
- The artefact's resistance to the technological changes;
- The reasonably priced integration of the artefact in the structure of the host system;
- The assurance of a reasonable reusability coefficient of the artefact during the structuring process of other artefacts;
- The flexibility of the relations that exist among the components of the artefact;
- The flexibility of the artefact's connections with the host system.

3. Preliminary remarks regarding the specification of the artefacts' behavior

The problem that concerns the specification of the artefacts' behaviour is solved in various ways by different modelling languages. Considering the software engineering from an UML perspective, it is immediate to notice the lack of an approach that is widely accepted in the industry, when speaking about the modelling of the behaviour.

The support that the UML offers concerning the behaviour is criticized as inadequate considering the demands of the CASE tools. More precisely, the code generation that is based on the UML specifications relative to the artefacts' behaviour is still unsatisfactory. In practice, the modelling becomes mandatory if the artefacts that are generated constitute a quality factor that is statistically proven, but also ingredients that may contribute to a better management of the IT projects. In fact, the fundamental reproach that is addressed to some UML artefacts, such as the sequence diagrams, as an example, is formulated in a simple manner: Are the costs that are incurred by the sequence diagrams realization justified? Many experts in the field of the software engineering argue that the answer is still negative.

The problematic of the behaviour's modelling will constitute the object of a future paper. Nevertheless, it is relevant to emphasize the close connection between structural and behavioural regarding the process of the software system's modelling. This close connection makes difficult for the boundary between static and dynamic to be drawn when the system is described. This paper

suggests that the transition from structure to behaviour is accomplished with the help of a special artefact, the interface. It can be considered ambivalent, as it features both structural and behavioural affinities in connection to an artefact.

4. Acknowledgements

This work is supported by the Transylvania University of Brasov, Romania.

References

- Bocu, D. & Bocu, R. (2013). Remarks on Interface Oriented Software Systems Modelling: International Journal of Computers, Communication and Control, Vol. 8, No. 5.
- Meyer, B. (1997). Object-Oriented Software Construction: Prentice Hall., 2, 331-410.
- Pugh, K. (2006). Interface Oriented Design: The Pragmatic Programmers, 2006-7-23. Available at <http://buhoz.net/public/libros/design/Interface.Oriented.Design.Jun.2006.pdf>
- Saha, P. (2007) Handbook of Enterprise Systems Architecture in Practice: IGI Global, doi:10.4018/978-1-59904-189-6.ch001.
- Tidwell, J. (2010). Designing Interfaces: O'Reilly, December 2010: Second Edition, ISBN: 978-1-449-37970-4. Available at <http://internativa.com.br/mobile/Livro%20%20Designing%20Interfaces,%202nd%20Edition,%202010.pdf>