Object-oriented programming

Rafał Witkowski

2019

Mikołaj K

Poza tym nie wiem czy słyszeliście państwo o takiej technice jak programowanie obiektowe – chodzi z grubsza o to, ze obiekty którymi otacza się ludzi mają wpływ na nasze mózgi tak żebyśmy byli zniewoleni.

Programowanie obiektowe nawet wykorzystuje swoją specjalną technikę która nazywa się Jawa – przez to ludzie myślą ze symulacja na która patrzą to tak naprawdę jawa czyli jest prawdziwa

Przed chwilą   Lubię to!   Więcej

I don't know if you hear about the technique called object-oriented programming.

Basicly it is about that there are some objects around us that affect our brains so that we are enslaved.

Domain model

- The domain model aims to visualization of concepts (conceptual classes) occurring in the field in question (the context of the system being set up).

- The most important model is created during object-oriented analysis! Work need to be done at this stage is to identify the conceptual classes. Correctness the construction of this model guarantees success during the design and implementation phase.

- The disciplinary knowledge model is a visual representation of classes or real objects occuring in the real environment that we model. It is not a representation of programming entities, i.e. classes and objects written in the language programming.

Domain model – example

- In UML, the domain knowledge model is written using the class diagram it describes:
  - Real objects from a given domain or conceptual classes,
  - Associations between conceptual classes,
  - Attributes of conceptual classes.
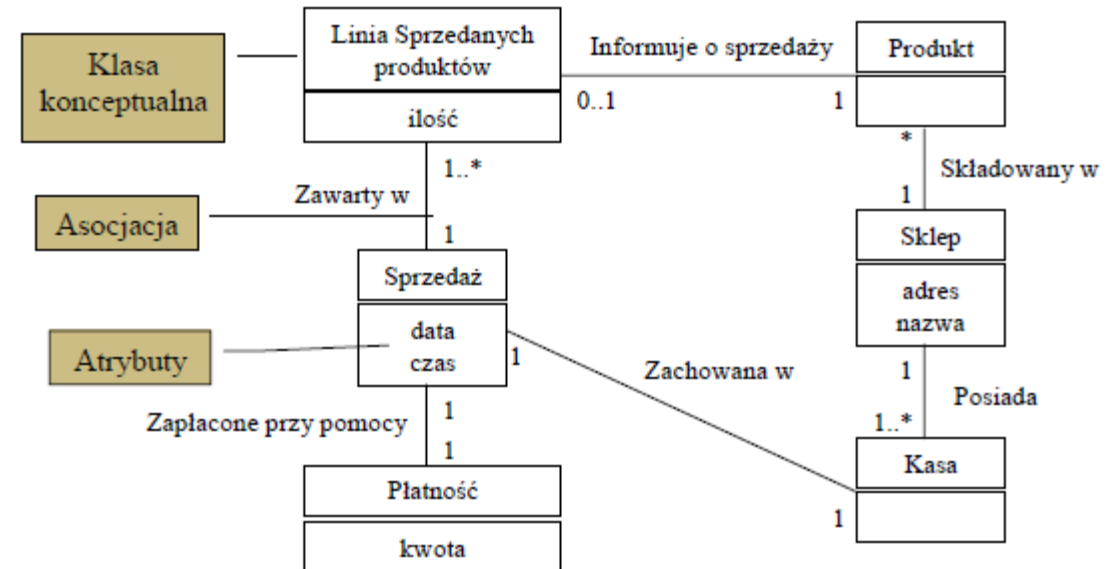
Domain model

- The domain knowledge model visualizes and combines conceptual classes occurring in a particular field.

- It refers to certain abstractions of conceptual classes because a given concept may represent different concepts depending on the context in question.

- Alternatively, concepts written by means of UML notation could be written using text in natural language, or a dictionary. However, visual language makes it easier to understand the concepts described and is an excellent way of communication knowledge about the created system among the team members.

- The domain model of knowledge in a given field can be treated as a visual dictionary of the relevant elements of the field under consideration.
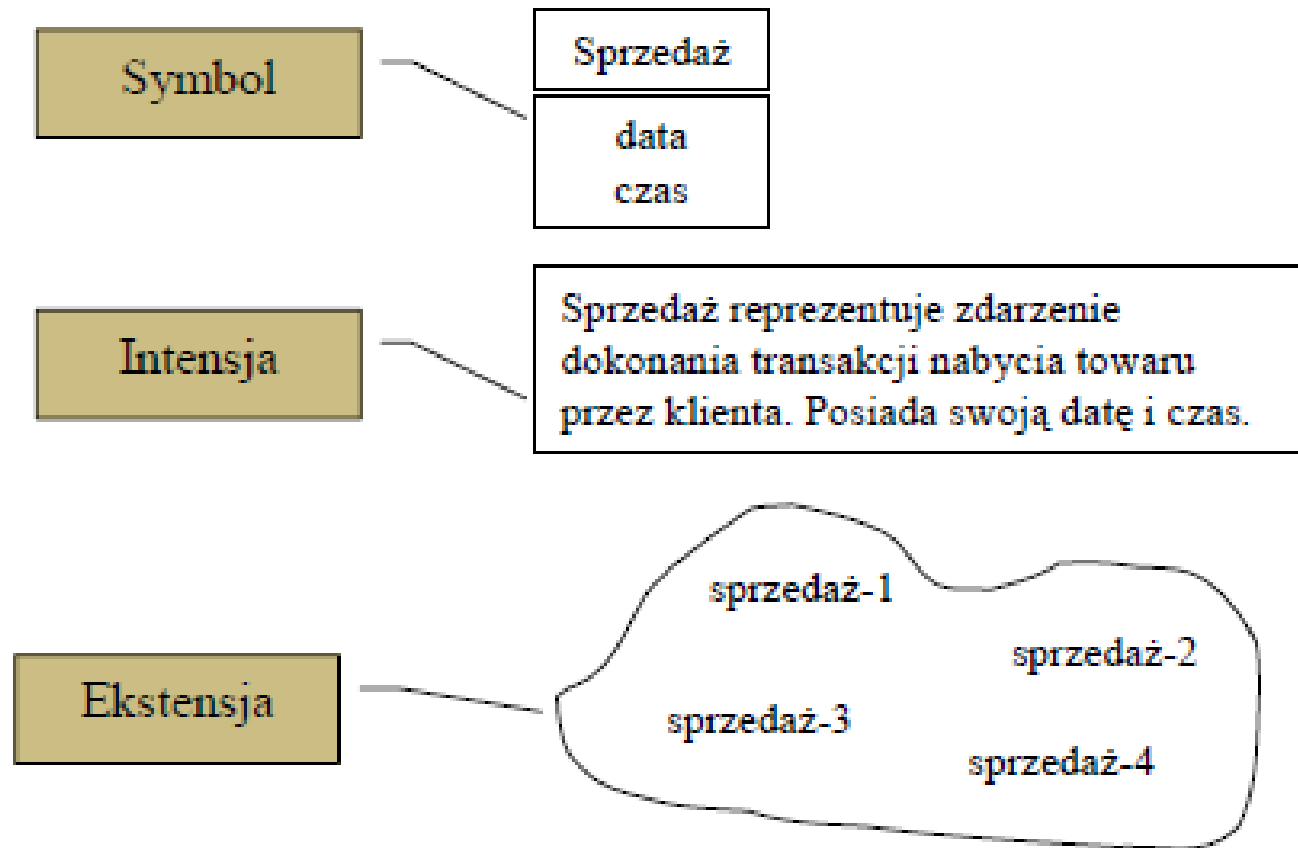
Domain model

- The domain knowledge model visualizes things from the real world, not classes written in programming languages such as C++ or Java.

- In models of this type you should avoid:
  - Programming entities such as the Window or Database,
  - Responsibility of objects and their methods.

Doamin model – creation process

- A conceptual class is an idea, a thing or an object.

- A conceptual class can be considered in three ways:
  - Symbol - a word or graphic element representing a conceptual class,
  - Intensity - definition of a conceptual class,
  - Extension - a set of examples to which a given conceptual class applies.

Domain model – creation

Identification of conceptual classes

- In an iterative process, the domain model of knowledge is not built in one step, but in stages corresponding to the scope of the work of each iteration.

- The identification is carried out on the basis of the results of the requirements phase (e.g. on the basis of use cases).

- It is assumed that for the identification of conceptual classes it is good to go beyond the specification of the knowledge model by the introduction of many (sometimes too many) very detailed measures conceptual classes rather than unspecifying the model.

- It is common to omit (because they are not found) in this phase of many conceptual classes, the subsequent phases are used to find them for entering attributes and relations.

- It is not recommended to exclude conceptual classes from the model only because they do not appear in the requirements or do not have attributes.

Identification of conceptual classes – categorization technique

What is looking for	Example
Organisations	SalesDepartment, Airline
Incidents	Sales, Payment, Meeting, Flight, Takeoff, Landing
Processes *(in most cases not represented as concepts)*	ProductSales, SeatBooking
Rules and principles of conduct	ProductReturnRule, ReservationCancellationRule
Catalogs	ProductCatalog, PartCatalog
Results of financial activities, effects of work, contracts	Invoice, EmploymentContract, ServiceAgreement
Financial instruments and services	CreditLine, Stock
Textbooks, documents, books	Operating Instructions, DailyListPriceChange

Identification of conceptual classes – categorization technique (2)

What is looking for	Example
Physical objects	Plane, car, cash
Specifications, designs or descriptions of things	ProductSpecification, FlightDescription
Places	Airport, Shop
Transaction	Sales, Reservation
Transaction position	SalesPosition
Role of people	Clert, pilot
Containers of other things	Outlet, basket, plane
Things in a container	Procuct, Pasenger
Information systems and electrical/mechanical systems outside the system	PaymentAutorisationSystem, AirTrafficControlSystem
Abstract concepts that can be described in nouns	Hunger, Agarophobia

Identification of conceptual classes: noun phrases identification technique

- The technique is based on the identification of nouns and noun phrases in the textual (oral) descriptions of the topic and to treat them as candidates for conceptual classes and attributes.

- The full description of the use cases is an excellent description of the topic which can be used as a basis for this technique.

- Because words in the language are ambiguous and different nouns (noun phrases) may mean the same concept, it is not possible mechanical creation of noun-class assignments. This technique requires a lot of attention!

- It is recommended that the technique of identifying noun phrases be used together with the technique of using a list of categories.

Identification of conceptual classes: Example: SalesSupport

- The use of category list and noun phrase identification techniques to analyse a use case SalesSupport can lead to the identification of the following candidates for conceptual classes :

Cashier	Customer	Manager	Sale
Product	ProductDescription	Sales	Item
Payment	CashRegister	ProductCatalogue	Shop

- There is no concept of a "correct" list!!!!! It is always a set of arbitrarily chosen terms by an analyst. However, using the same techniques, it should be obtained by various analysts from similar lists.

- The problem with recognizing concepts is important and introducing them to the model on the an example of a concept called Invoice:
  - Against: Repetition of information contained elsewhere in the model,
  - Pro: needed for a possible return of the product.

- Whether the concept is introduced or not, depends on iteration and whether the concept is relevant at this stage.

Domain model creation: recommended steps

- Step 1: Create a list of candidates for conceptual classes using the category list technique and the noun phrase identification technique based on the currently considerated part of the requirements for the system.

- Step 2: Save the results in the diagram.

- Step 3: Add the associations needed to describe the relationships between the concepts.

- Step 4: Add the attributes necessary to record the information obtained from the requirements analysis.

Domain model: strategy for knowledge dictionary use (cartographer's strategy)

- In creating a domain model of knowledge (cartographer's strategy, or a strategy for using a dictionary of knowledge):
  - Use the vocabulary of the topic when naming classes and attributes. For example, when modelling a library, we use the terms Lender and Librarian to describe the terms Client and Library Service Employee.
  - Remove conceptual classes from the model that are considered irrelevant from the point of view of the requirements for the system at a given stage of its creation. For example, let's exclude from the model the terms Pen and Advertisement as irrelevant from the point of view of the system requirements.
  - Do not introduce to the model entities that are not present in the discussed field.

Domain model creation: misktakes

- A common mistake is representing existence as an attribute instead of a conceptual class. In order to eliminate this problem, it is recommended to use a rule:
    - If we do not think of a potential conceptual class X as a number or text in the real world, then X is probably a conceptual class, not an attribute.
- In case of doubt, in most cases we introduce an independent conceptual class.

Domain model creation: „unreal world" modeling

- Many IT systems concern fields that have no direct connection with the real world, e.g. software developed in telecommunications.

- For such applications it is possible to create domain models of knowledge, but this requires a high level of abstraction and use analogies from fields for which modelling has already been done.

- For example, in telecommunications topics, the terms Message, Connection, Port, Dialogue, Road, Protocol can be used.

Reducing the representation gap

- The domain model provides us with a visual representation of a dictionary. We should draw on the model of this dictionary inspiration for naming programming entities during the phase design and implementation.

- Such an approach allows to reduce the representation gap (gap semantic) between our (i.e. analyst) model and its representation in a specific topic, and its representation in a specific topic – an IT solution.

- At the level of the domain model we use certain concepts (e.g. Sales), at the level of entity design Programming (Sales class). They are not the same entities, but the second being was created by being inspired by the first.

- This is one of the main advantages of the object-oriented approach!!!!

Example of domain model phase