The Ministry of Education and Science of the Russian Federation
Samara State Aerospace University
(National Research University)

N. M. BORGEST

ONTOLOGY OF DESIGNING

Electronic Textbook

Part 1
Concepts and Principles

SAMARA

2011

УДК 004.9(075)
ББК 32.97

Author: **Borgest Nikolay M.**

Translated by: **Borgest Diana N.**

Borgest, N. M. Ontology of Designing = Онтология проектирования [Electronic resource] : Electronic Textbook / N. M. Borgest; The Ministry of Education and Science of the Russian Federation, Samara State Aerospace University. - Electronic text and graphic data (3Mb). - Samara, 2011. - 1 CD-ROM.

This textbook is a part of the Masters educational programmes which were developed based on using new educational technologies, resources and distance-learning systems for the Masters programme «Designing, construction and CALS-technologies in Aeronautical Engineering » for education direction 160100.68 «Aeronautical Engineering».

Prepared by the Department of Aeronautical Engineering SSAU.

CONTENTS

FOREWORD

The module “Ontology of Designing” for the Masters programme “Designing, construction and CALS-technologies in aircraft engineering” includes:

this textbook;

guidelines for laboratory works [6];

scientific articles [2];

recommended sources [3].

The main part of the module, which most probably will be memorable for the students and will accompany them in the future, is **ontological analysis**. Semantic analysis is a component part of the ontological analysis. Applying it to the above-mentioned components of the educational content, students should pay attention to the meaning and significance of these structured components.

The textbook is the core of the educational content. The given material cannot teach itself. It helps to learn the subject. The aims of the educational component are to give direction, to define a range of literature and the basic glossary of the studied field, to reveal the essence of the used methods, theories and hypotheses, highlight the problems, and to indicate possible ways of solving them.

The guidelines, which complement the textbook, show how to do the practical works. Under this module it is offered to do a range of laboratory researches solving design problems with a use of ontological systems.

The scientific articles indicate the results of real researches in the domain as well as show the depth of solvable problems. They will be useful to fully understand the subject.

The list of recommended literature and internet resources complement the educational content. One can find it at the end of the textbook.

Two worlds dominate from the century,
Two equal beings:
One embraces human,
Another is my soul and thoughts.
And as in a tiny dewdrop
You recognise entire face of the sun,
So together in the cherished depths
You will find the Universe.

Afanasy Fet (1820-1892)

INTRODUCTION

Replacing some words in the famous phrase about a civil position of a poet it will look as 'You don't have to be a Master, but you do have to be an engineer'. Or with another ending '…but you have to defend (your research – Прим. пер)'. Not all the students will become researchers, however everyone will have to design projects and explain taken decisions. For this reason it is important to see the main point, to isolate elements which form a developing project or a system, to know how to use systematic, object-orientated and process approaches, to see the relationship, to define functions, and to understand the goals.

The rapid development of information technology has significantly changed the process of designing and, as a consequence, of the requirements for narrow specialists who create complex systems and facilities. The simultaneous learning of information technologies and how an object is created (a plane, an engine, a spacecraft, etc.) often leads to a compromise in one way or another. Often when one is a professional at something, he or she has poor knowledge in another field. The increasing complexity of engineering and technology, natural and forced differentiation of engineering labour, multi-step design, duration and multi-stage life cycle of the product, and the large number of staff involved in the process of creating and operating the products, all these lead to a necessity of unambiguous interpretation the data and concepts of the domain. In spite of a significant part of the design procedures and operations that can easily be solved with modern computers, the most important areas are not computerised. An approach to solving these complex areas is seen as further intellectualisation of CAD.

The works in artificial intelligence have been going on for decades in different directions. On one hand, these works are about biological intelligence, its structure, evolution and the learning process. On the other hand, artificial intelligence is seen as an engineering discipline of computer simulation of intelligent behaviour in order to create the design.

Ontology is a traditionally philosophical discipline which found its development in ontological analysis. Ontology of designing defines the essence of the complex process of modelling the future. Ontology as a study of existence researches the laws that already exist. Designing is a creation of something that has not been made yet. However, in most cases the designing is based on studied and examined patterns. If in ontology of the material world there are examples of standard ontological models (e.g. Mendeleev's Periodic table of chemical elements), then "in ontology of non-material or social worlds there are no Mendeleevs yet. **It is**

just the beginning" [23]. Hence there are some difficulties in postulating "unambiguous" knowledge, and also a genuine interest which is common for all the pioneering works.

Thesaurus is the basis of many applied ontologies. It helps to organise the knowledge about the subject, to "negotiate" the understanding of the nature of symbols and concepts, and to define the terminology.

A quarter-century ago, when the author of this textbook had defended a thesis on optimising the parameters of the workflow of the aircraft engine came to work at the Aircraft Designing Department the first thing he started to do was dealing with terms. It was impossible to expand the boundaries of the domain without knowing and understanding the technical terms.

Five years later, in collaboration with colleagues of the Department "A short dictionary of aviation terms" was issued [4]. It has been a good help for aviation specialties students, where one can easily find the necessary interpretation of the term. Based on this dictionary a mini-wikipedia was created by a graduate of Moscow Aviation Institute (www.aviaslovar.ru).

In the first part of the textbook one can find a number of important terms according to the author’s opinion that will be useful for learning and understanding the discipline.

We will understand that we don't know anything after we die.
Avicenna (980-1037)

One who doesn't know anything is blessed because he can't be misunderstood.
Confucius (551-470 BC)

I know nothing except the fact of my ignorance.
Socrates (469-399 BC)

If I know that I know a little I will try to achieve to know more.
V.Lenin (1870-1924)

1. Terminological agreements

The amount of information has long gone beyond the critical threshold, so everyone feels amateur in almost all the subjects. The GOST (State Standard) Standard certification process of some terms does not keep up with the variety and diversity of interpretations of seemingly well-established concepts and newly introduced characters and symbols of various entities. Therefore, it is always important to agree on the meaning of terms, what they define, identify, what include, what is their content before the communication.

Some experts who introduce the systems, pass the data about the product through all stages of the life cycle are concerned about the same agreements. The new systems are piled upon the process, which is started by a person with only one purpose - <u>not to lose data and not to be confused in it</u>.

Ontological analysis of the domain and the development of thesauri promote synergy, coherence, uniqueness, and thus efficiency and quality of decisions.

It is hard to disagree with great Descartes, who believed that "a half of misunderstandings" lies in the ambiguity of concepts (he was talking about accuracy of using a word). It is known that a correctly stated problem is already half solved.

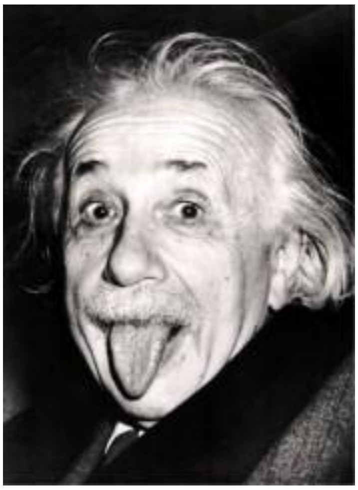
Based on what was written above it can be assumed that the reader, who thoughtfully and carefully read this section, has already mastered half of the subject. Congratulations!

The discipline's important terms (such as designing, optimisation, automation, formalisation, uncertainty, etc.) are also given in the chapter "Key Terms" [1]. Despite the fact that the brochure was issued some decades ago, it is still important because the given material, including the concept of intellectual CAD has not been implemented neither in our country nor abroad. Professor O.S. Samoilovich when read the manual he said that our works on the intellectualisation of the initial phase of designing an aircraft (of course, in those "distant" times) were leading. Therefore, we strongly recommend you to read this brochure; it will add significantly to the material on ontology of designing presented below.

The already mentioned “A short dictionary of aviation terms” [4] would be a good addition to the personal vocabulary, as well as its electronic version at www.aviaslovar.ru.

Artefact (Latin *Artefactum* – “artificially made”) in general understanding is any artificially created object or any phenomenon that does not exist in nature in native form. It is also something that has been created by man especially an object of archaeological or historical interest. In other words, A. is an ancient thing that is of particular value, or has a special meaning. In biology, A. is an unusual for a healthy body process or a formation.

God (creator) is an absolute of knowledge. The idea of God is interpreted in different religious traditions of humanity as omnipotent higher being that creates the world and gives the value, the meaning, and the law to the things and creatures. Being aware of our imperfection we measure ourselves with the idea of the all-perfect being. This idea is not our invention, as well as it didn’t derive from experience. It could be put into us only by the all-perfect being. Complete perfection is possible only under the condition of possessing all the features at the highest level, and, consequently, the full reality, which is infinitely superior to our own reality. Descartes’s proof of God's existence is a combination of anthropological (psychological) and ontological points of view. In our case, the introduction of the concept of the Absolute and the virtual all-perfect being is needed for ontological clarity, integrity and completeness of the process, as well as for understanding the essence of natural and artificial selection, place and role of the subjects of designing. The Absolute as an immaterial entity and a virtual "owner" of knowledge of the universe gradually reveals it to its intelligent creatures.



Devil is the Biblical antithesis of God. It is traditionally associated with evil, destruction, and a suffering place (Hell). Therefore, not all knowledge that is open for a man is good for him. Everyone knows where the road with "good intentions" leads. Wise use of knowledge is related to moral and ethical qualities of the *subjects*. Knowledge is power; and it is important which hands hold this power. Such inventions as Garin’s Hyperboloid, the hydrogen bomb, the Hadron Collider are all works and thoughts of a human. Everyone decides how to use this knowledge. Leonardo da Vinci, Nikola Tesla, Albert Einstein tried to correlate the need and desire for new knowledge to readiness of humankind to use it with its own benefits. All in good time, and as it is known time is relative.

Ontological time is a change. It is the unity of past, present and future. There is no time without changes. Traditional physical time is a measured movement. Psychological time, on the contrary, is individual and subjective; it is nothing other than a state of mind. Time is not static; it moves and has a direction. For each single subject present time is a unique egocentric temporal point, thanks to which its future and past can be seen. Ontological point of "the end" means that there is no present, no future, but there is a past, which can be preserved with the interpretation of the experience. Of course, if there is anything to be interpreted [23.]

Time as a philosophical category and an attribute of ontology, really affects the parameters of the designed system and decisions. Therefore, it must "be presented" in the designed object [1].

Temporal principle of optimisation. Improvement of the characteristics of optimised subject should be implemented only by changing the design variables. In order to get a valid result of optimisation it is essential to sustain the conditions that will ensure no impact on the results of other factors besides the varied parameters. In particular, in optimisation of the subject of designing it is important to keep the eternal perfection of the properties of future design (e.g., weight, design, technological, thermodynamic, aerodynamic), which should correspond to the period of time when the serial product release is planned [1].

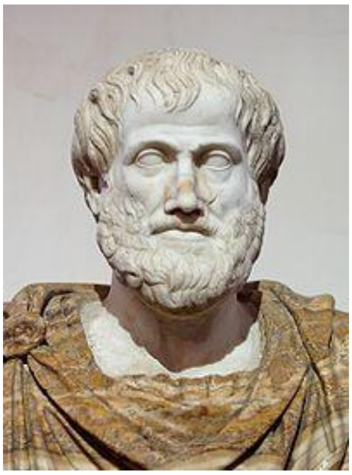
Temporal principle of designing. The designing of a subject should be done at a certain time interval, considering technological, scientific and technological achievements that are expected in this period. The famous maxim “If you want to make God laugh, tell him about your plans” should not be an obstacle. One of the fundamental features of human existence is the planning of human activity and its results. The desire to look into the future based on the adaptation of social and organisational experience. It is directly related to the prediction of performance [23].

Gestalt psychology (German *Gestalt* - "essence or shape of an entity's complete form") was founded by Max Wertheimer in the beginning of the 20th century. His key work was “Productive Thinking”. "Thinking lies in envisaging, realising structural features and structural requirements; proceeding in accordance with, and determined by, these requirements; thereby changing the situation in the direction of structural improvements...”. The basic postulate says that the integral structures (gestalts) are the primary data of psychology. Gestalt conveys the idea of an integral whole. The simplest form of perception is to separate the visual sensations into the object – a *shape*, located on the *background*. The Gestalt approach is used to create applied ontologies [30]. The main hypothesis can be stated as: “Harmony = conceptual balance + clarity”.

Glossary (Latin *Glossarium* – “dictionary”) is a dictionary, which consists of specialised terms in any field. Each term in the glossary has detailed descriptions and examples.

Project Life Cycle is different to the Product life cycle. It determines the sequence of phases of the project based on the needs of project management. Project Life Cycle has five phases: Initiating, Planning, Executing, Controlling and Monitoring, and Closing.

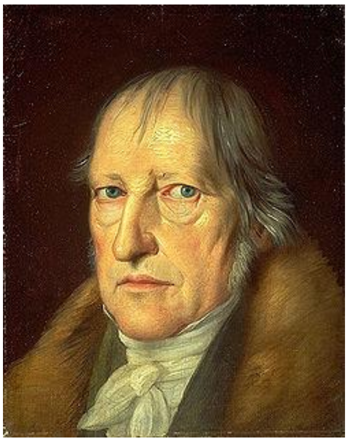
Logistics, in the traditional sense, is a part of economic science. It includes organising and regulating the flow of goods and services between the point of origin and the point of consumption. It manages the distribution of products, goods, services, as well as the stock. In our case, logistics contains dynamic and static conceptual models which describe the reality, logic and sequence of various tasks, including business needs. The model of logistics is an ontological pattern, one of the tools for describing, structuring and articulating the experience of the enterprise.



Metaphysics, or first philosophy, is a speculative doctrine of the fundamental nature of being and the world. The word *metaphysics* derived accidently. When the students of Aristotle were organising all his works, a number of books with the arguments about the origins, were placed after tracts in physics and were signed as followed by physics (books) - meta physics. It was understood in a figurative sense as the very content of "first philosophy" (by Aristotle); the word metaphysics points to the study of what lies beyond the physical phenomena. This interpretation of the term has remained in general understanding.

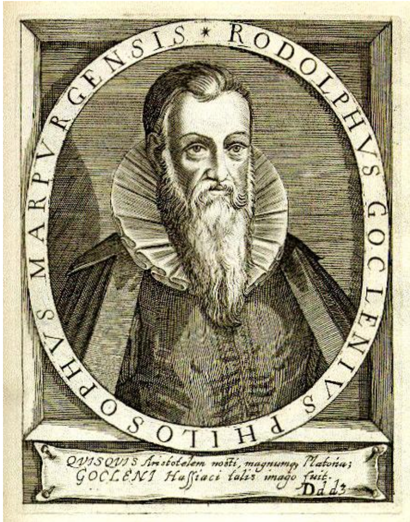
Monad (Greek *Μονάς* – “unit”, “a simple being”) is an invisible metaphysical unit, which carries some information. **Monadology**, Gottfried Leibniz’s work of 1714, describes monads.

Mutation (Latin *Mutare* – “change”) is a multivalued term. In biology, it is a persistent change in a genomic sequence, occurring under the influence of external or internal environment. The process of mutation is called mutagenesis. In physiology, voice mutations are age-related changes of human voice. Mutation, in music theory, is a compositional technique, which explains the expansion of the sound system. In creation of artefacts, it means a change of the object properties by changing its structure (in aviation, the creation and application of new materials, engines, avionics, etc.).



Objects of designing are real or virtual entities developed by the subjects of designing. These include physical facilities, systems, machines, equipment, computers, and software.

The term ***ONTOLOGY*** *is the most important of the discipline. Several interpretations of the term from various sources are given for clarity and better understanding.*



Ontology (from the Greek *on* (*ontos*) – “of that which is”, and –logia – “science, study, theory”) (Encyclopaedic Dictionary of Philosophy. 1998) is a study of being. Since the early 17th century, ontology was considered as metaphysics of being and things, which are the foundation of metaphysics in general. According to Hegel, ontology is just the "doctrine about abstract definitions of the essence." In the 20th century ontology was again revived. The difference between the old and modern forms of ontology is that the first one considered the whole world in its relation to Man. This makes Man the ultimate goal of world order. The new ontology has developed a broad concept of reality, giving the full reality to the spirit. From this position it tried to determine the independent existence of spirit and its activity in relation to an autonomous existence of outside world. The old ontology limited the real world by material.

Ontology (Brockhaus-Efron Encyclopaedic Dictionary) is a general theory of existence. Such notions as *something* and *nothing*, *possible* and *impossible*, *definite (*or *real)* and *undefined, quantity, measure, quality*, *order* and *the truth* (it is formally defined as the unity of diverse, or the agreement of difference), as well as space, time, movement, shape are considered there. A study of simple units of real life, or the monads, is stated.

Ontology (Soviet Encyclopaedic Dictionary, SES) is a branch of philosophy, a theory of being, which investigates the universal basis, the principles of being, its structure and laws.

Ontology (Ozhegov’s Dictionary) is a philosophical study of general categories and laws of being, that is in unity with the theory of knowledge and logic.

Ontology (Great Encyclopaedic Dictionary, BES) (from the Greek *Ontos* – “real”) is a branch of philosophy, a theory of being (as opposed to epistemology which is a study of cognition), which investigates the universal basis, the principles of being, its structure and laws. The term was coined by the German philosopher R. Goclenius (1613).

Ontology (Dal’s Dictionary) is a study of the nature or entity, being, essence.

Ontology (in computer science) is an attempt to formalise certain knowledge with a conceptual scheme. Usually such scheme consists of a data structure containing all the *relevant* object classes, their relationships and rules (theorems, limits) adopted in this field. In computer science this term derived from "ontology", an ancient philosophical concept. In artificial intelligence, ontology is an explicit (clear, open) specification of conceptualisation.

Enterprise ontology is a formalised description of domain knowledge of each individual company. Ontology describes the object classes (such as business assets of the company, projects, tasks, orders and plans) and their attributes, relationships and processes. Ontology allows introducing changes to the system without specific programming skills and disrupting the system.

Ontological model of enterprise is an integrated model of a particular developing enterprise, which allows structured describing of its activities as a single unit. It provides self-management, survival and adaptive behaviour in life environment during the entire life cycle.

Ontological analysis separates the real world into components and classes of objects. It identifies their ontology, or a set of fundamental properties that determine their behaviour and change.

Ontological analysis usually starts with creating dictionary of terms and a system of precise definitions of these terms. The dictionary is used in discussing and studying the characteristics of objects and processes that constitute the system. In addition, the basic logical relationship between the introduced terms is recorded correspondingly.

There are two main categories of objects of perception in any system, such as the objects themselves, which form the system (physical and intellectual) and the relationship between the objects which characterise the state of the system. In terms of ontology, the notion of the relationship clearly describes the dependence between objects in the real world, and the terms are accurate descriptors of the objects themselves.

According to **IDEF5** method, the process of constructing ontology consists of five basic steps:

1. **Organising and Scoping**. This activity involves establishing the purpose, viewpoint, and context for the ontology development project and assigning roles to the team members.
2. **Data Collection**. This activity involves acquiring the raw data needed for ontology development.
3. **Data Analysis**. This activity involves analysing the data to facilitate ontology extraction.
4. **Initial Ontology Development**. This activity involves developing a preliminary ontology from the acquired data.
5. **Ontology Refinement and Validation**.

Ontological patterns are the tools for describing, structuring and articulation human experience. It is an individual means of typological fixation of life for each single subject. They describe a complete episode, a situation in its essential features and properties related to a specific social entity, in schemes.

Designing Patterns are repeatable architectural designs, which are the solutions to the designing problems within a frequently occurring context.

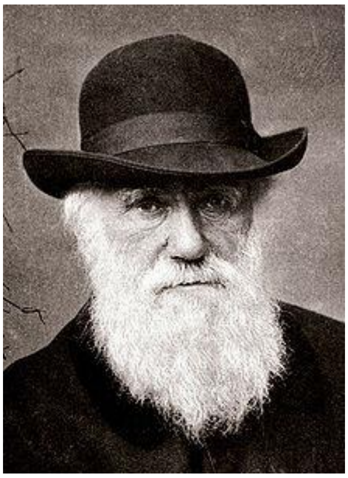
Usually a pattern is not a finished model; it's just an example of task solution that can be used in different situations. Object-oriented patterns show relations between classes or objects without specifying which final classes or objects of application to be used.

Algorithms are also patterns, but of calculations, as they solve computational problems.

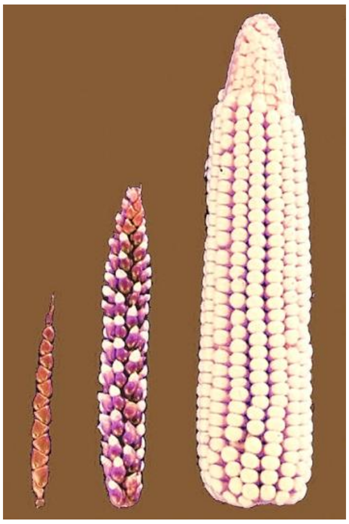
Ontology of designing is a formalised description of knowledge of the **subjects of designing** about **the process of designing** new or upgrading **artefacts**, including knowledge about the **object of designing** and similar to it artefacts, as well as a **thesaurus** of the domain.

Designing Experience is a personified observation accumulated for a specific period of time by the subject of designing during specific task solutions under specific circumstances. It is organised, structured, recorded and preserved in the system memory in the form of *ontological patterns* and unique for each subject. Experience is similar to organic synthesis; it is the integration of disparate episodes and situations by reduction the activity itself with keeping the existing [23].

Natural selection is the process that leads to survival and reproduction of more adapted to these environmental conditions creatures with useful traits inherited. In accordance with Darwin's theory and the modern synthetic theory of evolution, mutations, random hereditary changes, and combinations thereof are the basic materials for natural selection. An entity, conventionally called God, is the main creator and controller of these processes. Man, as one of His creations, has an honour to create within his own perfection and understanding. This is also the achievement of natural selection and the Absolute's decision. Artificial selection is delegated to Man by the Creator. However, it is a product of the Absolute's natural selection. The current process of designing artefacts and their modification and improvement done by Man are just a pitiful copy of what has been made by the Creator. But understanding this life is no less interesting and beautiful!



Artificial selection is the process of creating new breeds, varieties of cultivated plants and artefacts. It is done through the systematic preservation of animals and products with specific, valuable to the human traits and characteristics. This explanation broadly interprets the common concept. Darwin identified two forms of artificial selection: unconscious and conscious, or methodical.



Methodical selection is that which guides a man who systematically endeavours to modify a breed according to some predetermined standard. A task is deliberately set by a breeder who carefully selects manufacturers that correspond to the target. The rest is eliminated. Therefore, the goal is achieved by the choice of the best and the removal (elimination) of less relevant to the task.

Unconscious selection is an older form that existed in ancient times. Then Man did not set as a goal to improve their domesticated animals and plants. However, the best animals and seeds were preserved for breeding. Thus, less valuable animals and plants were destroyed, and used for food.

Darwin explained the cause of improving the forms and their diversity using the principle of selection. Sometimes similar requirements for various species could lead to identical characteristics in organisms of different origin. It is called convergence. According to Darwin, the large number of individuals is an important factor for the selection. In this case, the probability to find the necessary deviations

increases. The characteristics of artificial selection mentioned above are also presented in engineering. They are typical for designing the new objects with new properties (features). The selection of effective artefacts' features goes without interruption. Only the strongest survive!

A **need** is a kind of functional or psychological necessity or the lack of any object, subject, individual, social group, society. The needs vary from the situation. A need is a motive force for creating new artefacts and destroying old ones.

In psychology, it is common that the human needs are represented as a diagram or a pyramid. Physiological needs are at the bottom, and self-actualization is at the top. Abraham Maslow's hierarchy of human needs consists of the following levels

(upward):

1. Physiological; 2. Safety; 3. Love\Belonging; 4. Respect; 5. Cognition; 6. Aesthetic; 7. Self-actualisation.

In fact, the hierarchy places all the known biological species according to their stages of development – from lower forms to the Supreme essence. Not everyone here takes the place of honour. In the designing a need is transformed to a goal, and the goal to searching for solutions.

Subject area is a part of the real world that is discussed within this context. The context is, for example, a field of study that is a subject of an activity. It can be a designing as a form of research and development activity, or a designing of an aircraft as object-oriented methods. Subject area is also considered as a set of all subjects which properties and relations between them are regarded to a scientific theory.

An **enterprise** (in the context of the discipline it can be an engineering company) is an independent business unit. It is created to produce goods or provide services aiming to satisfy public needs and gain profits. Project and designing bureaus, companies and laboratories as social structures that solve project tasks also belong to the category of enterprises.

Project (from lat. Projectus "to throw something forward"). This term has two meanings. One defines it as a unique (as opposite to operations) activity that has a beginning and an end. It aims to achieve a predetermined result/goal, to create a specific, unique product or service with limited resources and time as well as with quality requirements and acceptable level of risk. According to the second one, project as a result of project activity, is a set of project documents that are needed to assess the future artefact and to make a decision of creating it.

External designing (macrodesigning) is a design process during which functional-structural issues of the whole system are solved. Macrodesigning begins with stating the problem of satisfying a need that includes: identifying a goal of creating a system and the range of tasks it solves; estimation of factors acting on the system and defining their characteristics; choosing the performance indicators of the system. The aims of the system are identified based on needs of practical application and taking into account technological change. Having experience in application of similar systems is very important.

External designing includes:

Conceptual design, which includes planning of concept and technical specification. Pilot project is the final document on this stage.

Draft design includes development of preliminary project decisions on the system and its parts. *Draft project* is the final document of the works at this stage. It contains design decisions of the subject of development as well as data that defines its purpose and the basic parameters.

Internal designing (microdesigning) develops elements of the system as physical units of equipment.

Internal designing includes:

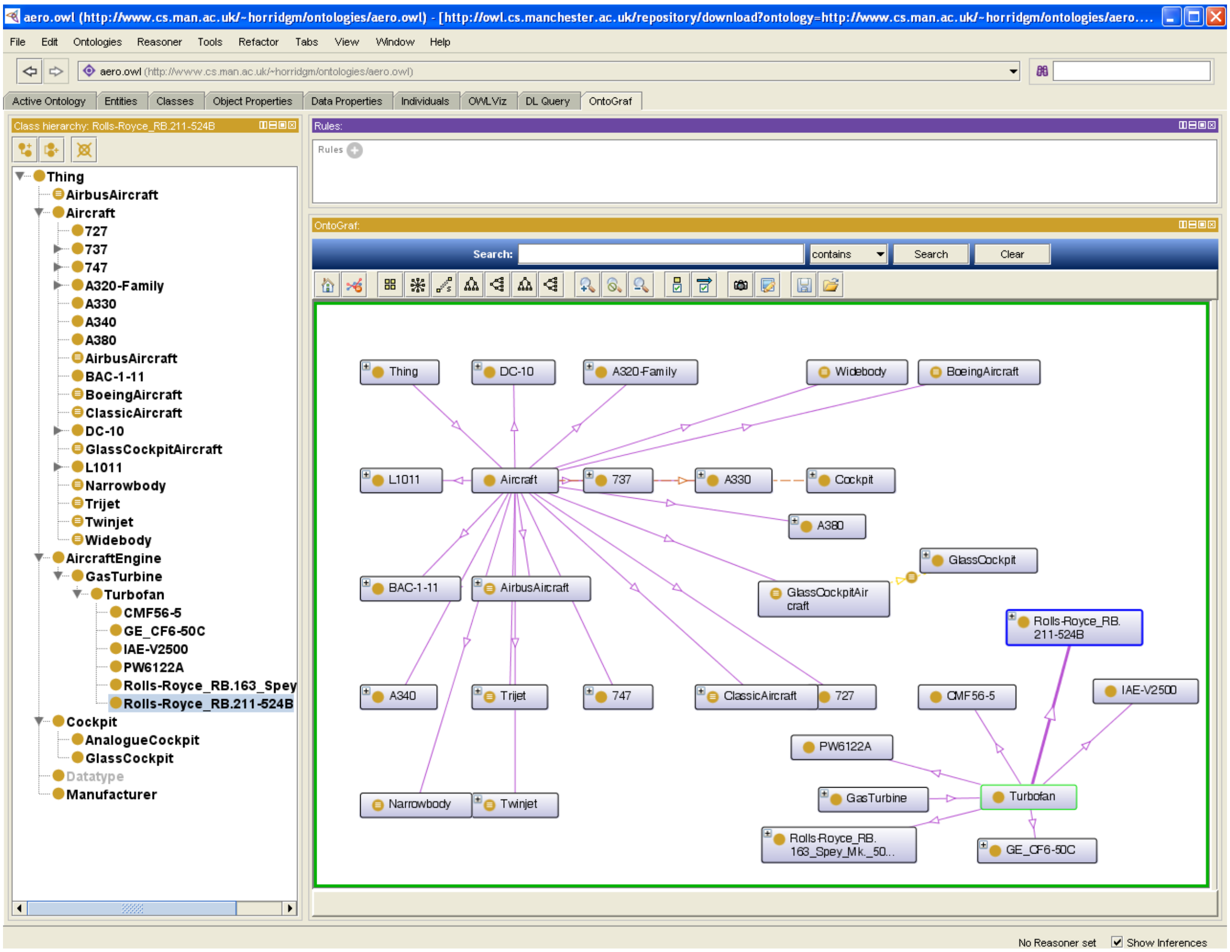
Preliminary design that includes development of project decisions on the system and its parts; development and preparation of documents for the supply of products for the acquisition and/or providing technical requirements to design the items; preparing designing tasks for the adjacent parts of the project. *Technical project* is the final document at this stage. Apart from described above materials, it contains fundamental schemes and design documentation of the designing subject and its parts.

Detailed design is the final stage of designing. It includes development of working documentation on the system and its parts as well as clarifying the results of the previous stages, designing and testing a development prototype and preparing technological and operational documentation. The results are given in the *working project*.

Project task (conditional ontological idealisation) is a well-structured understanding and description of the objectives of the activity related to a particular part of the activity of a specific subject of designing under specific conditions. It is characterised by the use of certain methods in a given environment which provides the achievement of this result in a definite time. All task classifications begin with the "clear and unambiguous personification. The term latently assumes those who will carry out this task". [23] The tasks are divided into standard (routine) and creative (non-standard). The presence or lack of experience of this particular subject of designing is the criterion of the typological division. Accumulated ***ontological patterns*** can be extracted and used in order to solve standard tasks, and be translated for new basic data and conditions. Considered as primary, non-standard tasks are related to analysis of the singularity, and thus they provoke thinking. [23] Therefore, it is important to compose a catalogue with the description of all standard and non-standard tasks and their solutions.

Space belongs to the category of ontological coordinates of the ontological model and includes at least two components: global geography and infrastructural space. Global geography is a typological structuring of physical space where subjects of designing (in this case) operate and interact with the environment and other subjects. Infrastructural space is a typologically organised structure of the design environment. It provides transparent and efficient management and control in tasks, departments and processes. Infrastructural space is specific for a particular company and cannot be automatically transferred to another one.

Protégé is a human under patronage or protection of a powerful person. It is commonly used referring to a gifted person who receives help in promotion from a more experienced and rich well-wisher. According to the designers from Stanford University, their software, a famous ontology editor **Protégé**, possesses similar features.



The **design process** includes target-setting, clarification and analysis of the design situation, divergence (expansion of boundaries), transformation (changing of the form), convergence, harmonisation (elimination of conflicts and contradictions), expert evaluation, and protection.

Target-setting methods: coercion (forceful imposition of a goal); agitation (brainwashing); a "free choice" (from a limited number of possibilities); a systematic approach.

Clarification of the design situation methods: interviewing; questionnaire; survey of consumer behaviour, system testing.

Methods of analysis: structural; functional; conflicting; causal; streaming; Cost Method.

Divergence methods: Elmer Gates' Method; Haefele's Notebook Method; Brainstorming developed by Alex Osborn; Synectics by William Gordon; Method of focal objects by Charles Whiting; Fritz Zwicky's Morphological Analysis; Method for increasing ideality.

Convergence methods: the Method of canonical criteria; beauty formula; Ranking Method.

Methods of harmonisation: conflict resolution principles and techniques; standard solutions of inventive problems; the Algorithm for Inventive Problem Solving (ARIZ); harmonisation algorithm.

Transformation methods: amplification; miniaturisation; disproportion; multiplication; deformation.

Expert evaluation methods: Peer Review Method; simulation.

Protection methods: creation and demonstration of the prototype; the defence of the project before the expert board.

Methods of organising the design process: Functional Cost Analysis (FCA); quality circles; 8D (Global 8D, Ford 8D, or TOPS 8D); FMEA; PDCA.

Engineering psychology is one of the special disciplines of psychology. It solves such problems as: 1) the rational organisation of human activity in the systems "Man and Machine" which are designed for managing and processing the data, 2) efficient allocation of functions among managers, service personnel and technical means of automation, and 3) optimisation of data support and decision-taking. Engineering psychology is based on personality psychology, psychology of labour, etc., and also closely cooperates with **systems engineering** and other engineering disciplines.

Design psychology refers to applied psychology. Engineering psychology, cognitive psychology (virtual psychology as its new subdiscipline), and acmeology (it is the science of regularities and ways to achieve excellence in all types of individual human activity) are considered the closest disciplines to design psychology.

There are always ***subjects*** and ***objects*** of designing. Subject sets the goal, defines the methods and means, as well as selects resources to create a specific artefact, an object of designing. The personal characteristics such as preferences, predilection, perception, understanding, experience, form ***a need*** and estimation criteria of the future object of designing as well as define agents with their methods, subcontractors and their methods, consumers, and the market. Human factor plays a great role in the process which starts with an idea until the final costumer. As opposed to engineering and virtual psychology, design psychology, creating a model of human creativity, studies the process of formation and transformation of the needs into an imaginary virtual object.

Revolution (from the Latin *Revolutio* - "rotation") is a fundamental change in power or organisational structures that takes place in a relatively short period of time. Revolution is opposed to evolution, which means a process of a gradual development. There is a common understanding of revolution in social structure, ideas, minds, literature, and science (e.g. Darwin was a revolutionary in biology, Kant in philosophy, etc.).

In document retrieval, **relevance** (from the Latin *Relevo* - "to ease") is a semantic match of a search query and a document representation; it is a measure of suitability between the obtained documents and those that the system should provide. ***Adequacy*** is the closest term to *relevance*, i.e. an evaluation of compliance and practical applicability of the results.

Semantics (from Greek, “sign”) is the study of meaning of words, phrases, and symbols that are used to transmit information. It focuses on the relation between signifiers and what they stand for.

A **Semantic Web** is a network which represents semantic relations among concepts. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges. It also consists of arcs and nodes which can be organised into a hierarchy. Concepts, events, properties, and processes can be the objects. A Semantic Web is one of the ways of representing knowledge.

The ISA relationship is the relationship between an object and a multitude, signifying that the object belongs to the multitude. The ISA relationship suggests that the properties of the object are *inherited* from the multitude.

The relationship between a superset and a subset is called AKO, “a kind of”. For example, “a plane is an aircraft” = a “plane is ‘*a kind of*’ aircraft”. The elements of subset/superset are respectively called hyponym (a plane)/hyperonym (an aircraft).

An object usually composed of several parts, or elements. For example, a plane consists of an airframe, an engine, avionics, etc. **HasPart** relation represents part-whole relationship (meronymic relations). *Meronym* is a part of a whole. An engine, for example, is the meronym for a car or an aircraft. *Holonym* means a word that names the whole of which a given word is a part. For example, a sailplane has a wing; so a sailplane is a holonym of a wing. A wing is a holonym of a longeron.

In a semantic web it is often required to define the relationship between synonyms and antonyms. These relations are clearly duplicated in the network.

Systems engineering is an interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed over the life cycle of the project. Issues such as logistics, the coordination of different teams, and automatic control of machinery become more difficult when dealing with large, complex projects. Systems engineering deals with work-processes and tools to handle such projects, and it overlaps with both technical and human-centred disciplines such as control engineering, industrial engineering, organisational studies, and project management.

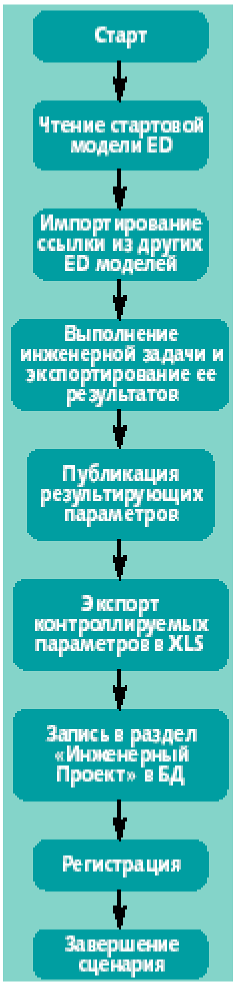
Design environment is an environment where new artefacts, the subjects of designing, are created. In addition to the subjects and objects of designing there is a variety of tools (other objects: real and virtual) and resources (material, intellectual, financial).

According to **GOST standards 2.103-68,** 2.118-73 (1995), the **stages of development** include technical proposal, draft, technical and working projects.

A *technical proposal* is a set of design documents that contain technical and economical explanations of feasibility of developing product documentation. It is based on analysis of the terms of reference, various possible product solutions, and comparative evaluation of solutions taking into account operational features of the developing and existing products.

A *draft project* is a set of design documents that contain fundamental design solutions. It gives a general idea about the structure and operating principles of a product as well as the data that defines the purpose, basic parameters and total dimensions of the developing product.

A *technical project* is a set of design documents that contain final technical solutions. It provides a complete idea about the structure of the developing product and raw data for the development of documentation.



Subjects of designing are rational beings involved in the process of creating an artefact, the **object of designing**. These include specialists (ontologist, cognitologist, psychologist, marketing manager, designer, constructor, engineer, economist, operator, executor,…), and God (the Creator, the Absolute). The subjects of designing have knowledge and also the ability to impart, gain and process it. God as the Creator of all things is also a subject of designing. His invisible presence is felt by ordinary creators who try to read the bits of knowledge about the universe, a small part of which is directed to improving the living environment and self-improvement.

A **scenario of designing** is the sequence of execution of project procedures and operations as well as a set of designing patterns. Consider a scenario of designing in terms of engineering design phase.

The work starts with uploading one of the starting models into the CATIA system from the SMARTEAM database. The starting model is an "empty" model which has predefined properties. For example, it has parameters, formulae, verifications, and other items of knowledge.

Then, a search for previously executed models (prototypes), which data is required for the developing model, is performed. If such models are found, then the process of importing links from them is carried out. Afterwards, the model is modified up to the required level. The results are published to become available for importing into other models (e.g., for using at the next stages of designing). Therefore, the design is confidential; and the models that inherit obtained results are "facilitated", which is important for the subsequent formation of large assemblies.

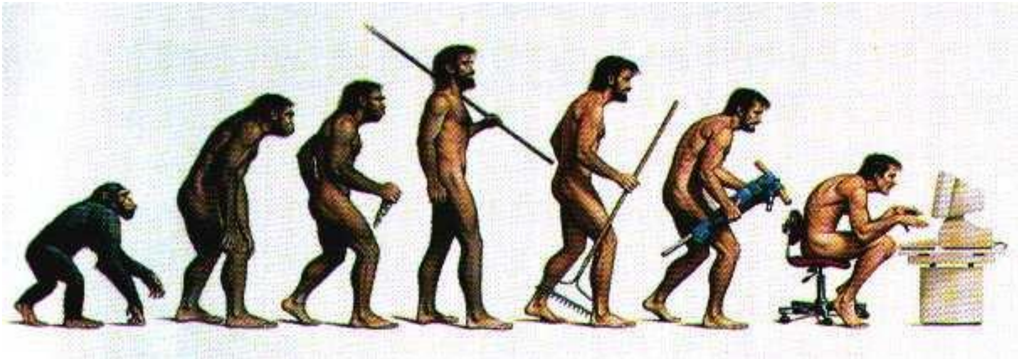
After that the model is checked in accordance to predetermined criteria. If the results are positive, then the model is validated and recorded in the SMARTEAM database. In fact, an automatic check for compliance to the rules is constantly done in the operating process in real time and not only at completion.

In modern linguistics, **thesaurus** (from the Greek *θησαυρός* - "treasure store") is a special kind of dictionary that lists words grouped together in accordance with similarity of meaning (synonyms, antonyms, paronyms, hyponyms, hyperonyms, etc.). Thus, thesauri, especially in electronic format are one of the most effective tools to describe one discipline, subject, or field of study.

In contrast to a dictionary, thesaurus entry does not define words but correlates a word with other concepts and groups. So it can be used for filling knowledge bases in Artificial Intelligence systems.

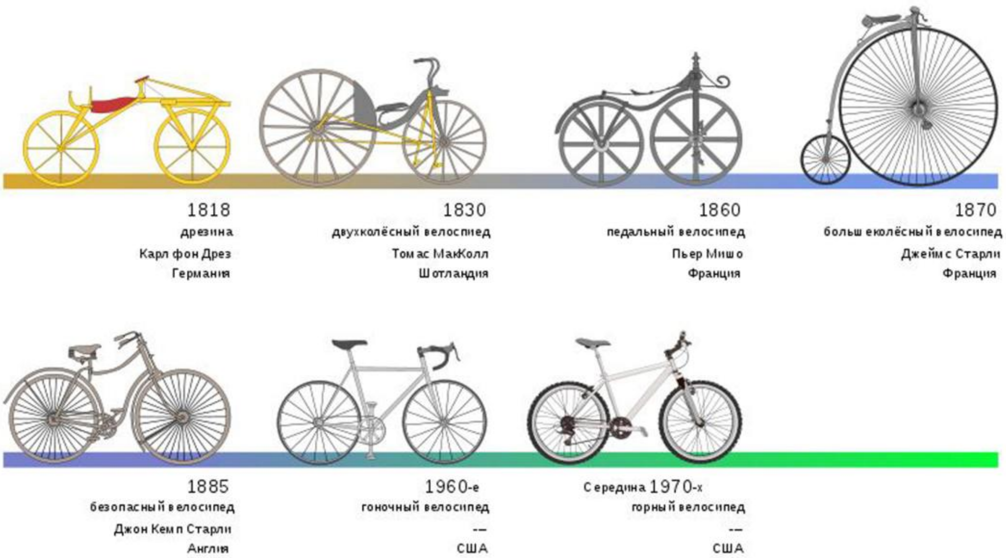
In the past, the term thesaurus mainly meant a “dictionary” with full description of lexis and examples.

A **frame** is an Artificial Intelligence data structure used to divide knowledge into substructures by representing “stereotyped situations”. The term “frame” was coined by Marvin Minsky in the 1970s for knowledge representation. A frame is an abstract model. It slightly describes the nature of an object, a phenomenon, an event, a situation, or a process. Usually a frame contains several slots; however, there can be just one. The slot is represented as (Name_slot, Value_slot).



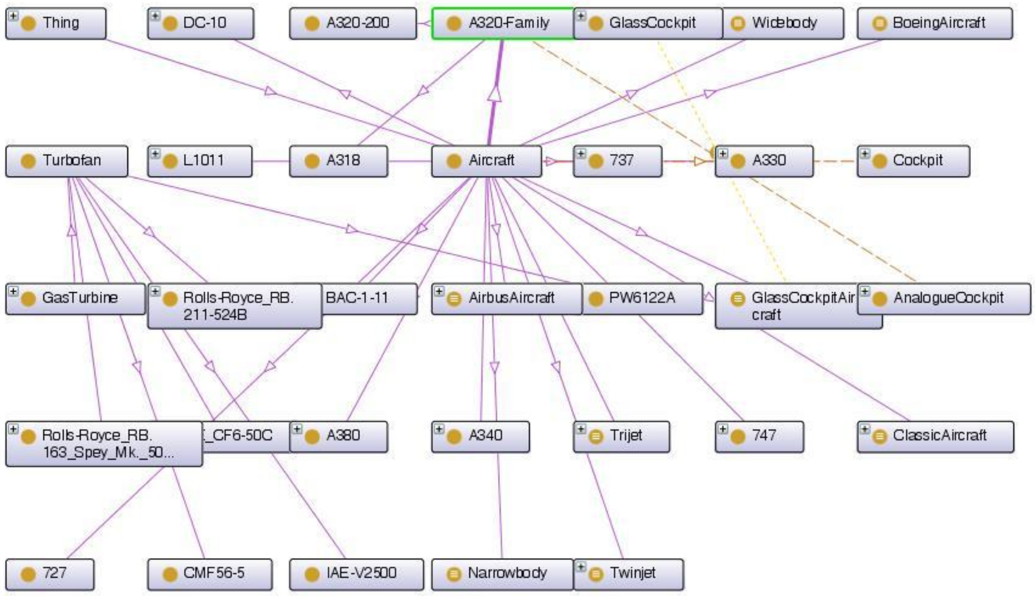
Evolution (from the Latin *Evolutio*, “unrolling”) is the process of a gradual change. The evolution of knowledge coincides with the evolution of life. In life as in the evolutionary process selection takes place. In the first minutes of the formation of the Universe, elementary particles were rapidly evolving. With the formation of chemical elements the process of evolution of minerals started. Minerals are parts of geological evolution and the evolution of organic matter. Biological evolution maintained the process inherited the existing structures and mechanisms of development. The evolution continued in the formation of communities of animals and human-beings. Man as a product of evolution, inherited evolutionary designing.

Academician A.N.Konovalov developed Engels’ ideas, expressed in the “Dialectics of Nature”, about the origin of life but based on already discovered supramolecular systems. According to the Academician, life is not a coincidence; it inevitably occurs there where elementary particles exist, which self-form atoms, atoms form molecules, molecules – supramolecular systems, supramolecular systems – biological systems... For this process special conditions are of course needed. Thank God, our planet has them. Yet.



Ontology Web Language OWL is a family of knowledge representation languages for authoring Web ontologies. Web ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms. An advantage of OWL ontologies is the availability of the tools that can provide generic support that is not specific to the particular subject domain.

IDEF5's ontology languages were created to support the ontology development process. There are two languages: the **IDEF5 schematic language (SL)** and the **IDEF5 elaboration language (EL)**. The schematic language is a graphical language, specifically tailored to enable domain experts to express the most common forms of ontological information. This enables average users both to input the basic information needed for a first-cut ontology and to augment or revise



existing ontologies with new information. The other language is the IDEF5 elaboration language, a structured textual language that allows detailed characterisation of the elements in the ontology. Various diagram types, or schematics, can be constructed in the IDEF5 Schematic Language. The purpose of these schematics, like that of any representation, is to represent information visually.

> If a painter should wish to unite a horse's neck to a human head, and spread a variety of plumage over limbs [of different animals] taken from every part [of nature], so that what is a beautiful woman in the upper part terminates unsightly in an ugly fish below; could you, my friends, refrain from laughter, were you admitted to such a sight?
>
> *Horace (65-8 BC)*

2. Engineering design. Artificial selection

Many were probably surprised by the fact that in the “Terminological agreements” chapter there were some terms that are not in common use among designers, such as artificial and natural selection, revolution, and evolution. So what is designing but an artificial selection! Moreover, in designing there are two forms of selection: unconscious and methodical. The changes that take place in the world of artefacts are of evolutionary and revolutionary nature.

In the very meaning of designing, methodical selection prevails in practice. At the same time, it is known, that the aviation pioneers did not expect and even did not dream that their “flying toys” made for fun, would be improved and form an industry that in a short term would change our world satisfying quite different needs.

In order to better understand the process of creating any artefact it is preferable to take a direct part in its creation. It is important to study the previous experience of designing similar objects. The subject “Aircraft” has many examples of how to create this complex technical system. The rich experience gained for over a 100-year period of aviation development in our country and abroad has not solved all the problems of designing. Neither the achievements in science nor the modern technologies, including CALS, cannot predict the characteristics of a future aircraft in high precision. Almost all estimates are approximate.

To illustrate the expressed assertion two examples of designing aircrafts are given below (in the U.S. and Russia). If we discard the difference in applied technologies, designing methods, types of engines and materials, and of course the way the material is presented, we can see that the results of modern designing (i.e. claimed parameters of an aircraft) are obtained with the same level of accuracy as 70 years ago.

SIMPLIFIED METHODS FOR DESIGNING AN AIRCRAFT

(the “Aviation” magazine, an article by R.R.Osborne, abridged
Source: K.Wood. Aircraft Design, transl. from English, Moscow, 1940)

Some see the process of design and manufacture of an aircraft as something mysterious. Many people do not have a clue as to why there are biplanes or monoplanes. We made a survey among some experienced designers on how they create a new plane. Below, the process of creating an ordinary aircraft from the drawing board to airdrome is given from their words.

Since the best draftsmen are busy with making commercials for the sales department, a chief designer discontentedly finds out that an inexperienced person will make the drawing, and he must do all calculations himself.

A designer is given a wingspan of 37.5 feet. However, a draftsman does not understand the handwriting and makes a drawing of a plane with a wing area of 375 square feet.

Originally, a monoplane was planned. Then, some district inspectors of the Department of Commerce were replaced; and a new inspector prefers biplanes. So, the design changes to biplane.

A golf partner of a chief engineer is the owner of a motor plant. This circumstance has a decisive influence on the choice of motor. The designer is outraged. The chief engineer begins to play golf worse; and the partner constantly beats him. Then he offers to the designer to take the best motor of another company. The designer is again puzzled: he does not know what to do with the new motor.

The president of JSC orders to reduce the cost of production and operational costs. The design is adjusted accordingly.

The designer finds out that an aircraft company X is designing a gull-wing plane. So, he immediately erases the drawing and starts to design a new gull-wing. At the same time a designer of company X erases the made drawings and decides to design a butterfly-wing as he found out that company Y is designing this type of wings.

The president of JSC returns from a trip around the country and sends a circular letter where he writes that improvements of the pilot view is the most important quality at the moment. So to achieve this goal the low cost and the speed are to be sacrificed. Therefore, the design is changed.

The production department shortens the fuselage by 1 foot by mistake. Before that, the production department covered the design bureau; so the designer wrote to the chief engineer proving that there is a trend toward a shorter fuselage; that is why the fuselage should be shortened by 1 foot. The chief engineer gives an order to reduce the fuselage nose by 1 foot. The designer and the head of the production department discuss this topic and decide to shorten the fuselage nose by 1 foot. The subject is considered closed.

Finally, an engine is arrived. It turns out, that the plant produced a 9-cylinder engine instead of a 7-cylinder one; and the under-engine frame was designed for a 7-cylinder engine. After a long correspondence between both companies about what to do, whether to change the under-engine frame or remove two cylinders, it is decided to throw a coin (heads or tails). The under-engine frame was replaced.

When the engine is installed, it turns out that the carbureter touches the chassis. The engine is sent back to the plant in order to remake the carbureter. When the engine is returned, it is discovered that the new carbureter touches the oil tank. The engine is again sent to the plant for modifications of the ignition system.

None of the fairing workers speaks English, so the design engineer explains them in layman’s terms what kind of wing fairings needs to be done. Thinking that he is talking about the engine hood, they make a new type of a hood for the engine.

The design engineer makes an appropriate drawing and sends it to the chief engineer with the indication that his new project should increase the speed by 4 miles per hour.

The chassis was designed for larger wheels. Someone made small-diameter wheels and sold them to an agent of the supplying firm. When the wheels were installed it turned out that the gap between the screw propeller and the ground is too small. The design engineer believes that a three-bladed propeller must be set as the blade tip speed is too high.

While assembling an aircraft it is discovered that the upper wing abuts the ceiling beam of the workshop. When the cost of the ceiling was compared to the price of one set of airplane racks it was decided to decrease the height of the biplane box by 6 inches.

After the first weighing it was discovered that the centre of the plane is strongly shifted. To obtain the required centre-of-gravity a new upper wing with a distinct sweep in the plan is produced. The chief engineer writes to the president of the company that the delay is justified by the improved pilot view.

When the plane is pulled out of the hangar doors, a left wing tip of the size of 1 foot gets broken off. The other side is shortened as well by one foot. Both edges are neatly rounded off.

The plane passes the speed test. The maximum speed is higher by 5 miles per hour than the designer expected but lower by 5 miles per hour than what he put in the preliminary specification. The speed is by 10 miles per hour higher than the one that engineering analyst expected and by 10 miles per hour lower than he promised to the president of the company. It is by 15 miles per hour higher than the sales department manager expected and by 15 miles per hour less than he specified in pre-advertising. That is the very speed the president of company expected to get as he knows well his business.

SUPERJET 100 IS FAR FROM CLAIMED PARAMETERS

2010, October, 26 “РБК daily” (abridged)

According to the materials of Sukhoi Company (GSC), for one and a half years of designing the jet SSJ-100/95B gained several tonnes. Thus, the maximum takeoff weight of aircraft has increased from 42.52 t to 45.88 t, and the maximum landing weight, from 39.4 t to 41 t. It caused greater fuel consumption. Before it was expected that when flying for up to 500 km the aircraft would consume 1651 kg of fuel, from 500 km to up 1000 km – 1615 kg, from 1000 km up to 1500 km – 1610 kg of fuel per hour, etc. The clients thought that the average fuel consumption is equal to 1622 kg. However, the manufacturer informed the first customers about the increase of the hourly average fuel consumption by 170 kg, so approximately up to 1800 kg per hour.

It is possible that the carriers would charge for extra expenses related to the operation of the aircraft. Thus, flying 3000 hours per year carrier’s additional expenses could reach 382.5 thousand dollars (with the fuel price 750 dollars per tonne). Rescheduling of the delivery dates of the jet could be another reason for an airline company to complain to the manufacturer.

The characteristics of a new aircraft are almost always exaggerated, then a programme for reducing weight begins. It also could be due to structural problems. But can a plane in this process achieve the declared figures? For example, the manufacturer of Boeing-787 left the first aircrafts for itself because the characteristics did not correspond to those stated initially.

The American newspaper “The Weekly Standard” published an article by Denis Manturov, Deputy Minister of Industry and Trade of the Russian Federation, which contained a detailed description of the major flaws in designing the aircraft engine (aircraft propulsion with two SaM146 engines supplied by PowerJet, a French-Russian joint venture between Snecma/SAFRAN consortium and NPO Saturn/Rybinsk Motors).

There have been repeated incidents of destruction of the disk of three-stage compressor. The first time this happened in 2009, then on the 9th of February, 2010, and the third case was in March, 2010, when the engine was tested.

However, the engine was certified as safe for passenger air transportation by the EASA (European Aviation Safety Agency). Therefore, the EASA certified a dangerous engine with fundamental design flaws.

Nevertheless, Sukhoi SuperJet-100 (SSJ-100) successfully completed the test programme to determine the noise level, which is a part of certification programme. Passing this test an aircraft is able to fly to Europe. This is important for SSJ-100 as it confirms good prospects in international market. At the moment, having the permission from IAC and EASA, the plane is taking part in tests in high intensity radiated fields (HIRF). The HIRF programme was developed by SSC with association of Alenia Aeronautica, a part of Finmeccanica concern, and other certification authorities.

Sukhoi Civil Arcraft CJSC was founded in 2000 to implement civil projects of the holding company. Net loss of Sukhoi Company in the first half of 2009 was 1170.5 mil roubles, that is seven times more than in the same period the previous year.

Now the company’s major project is designing a family of Russian regional planes Sukhoi SuperJet-100. The first flying aircraft was unveiled on the 26th of September, 2007 on Komsomolsk-on-Amur. On the 19th of May, 2008, it successfully completed its first flight. On the 24th of October, 2008, the plane was sent for certification after the plant’s flight tests were completed.

From the above articles it can be seen that in designing it is important to take into account all the factors (objects, subjects, their relationships and characteristics) that have influence on the effectiveness of future objects or systems. The first published work shows the power of human factor in terms of preferences and copying of ideas in design, lack of coordination between departments, and, as it may seem strange, low impact of decisions and values of selected parameters on the results of the designing (flatness of the functional).

The latter factor is explained with changing priorities in the process of design as the requirements and criteria are specified and formulated taking into account external and internal information. It resembles the well-known selection in biology when in nature there is a place for everyone according to their abilities and their needs. Above all is to survive, to take care of themselves. It is clear that an ***object*** cannot do it by itself, however the given by a subject resources can extend the life of a less perfect artefact. In some cases within the period more resources are needed to make a better object.

It is worth paying attention to the ***temporal principles of optimisation and designing*** that are not always realised in real practice of creation an object. The conditional idealisation of optimisation model, where only researched factors should be taken into account and the influence of the others is to be "frozen", is valid in terms of evaluating these factors. However, the non-project parameters which characterise the properties of the object and act as specified constants (for example, the properties of engineering material), cannot always be identified with a high degree of reliability.

In designing, the uncertainty of the project information as an absolutely objective factor due to natural reasons of predicting the future, is taken into account when searching for stable solutions to errors in forecasts. The problem of uncertainty of the project information, its nature and types is well described in the textbook [1]. There one can find information on design optimisation as the main procedure of the project.

Ecological principle is an actual principle of designing. The goals of the sons of God, designers of new systems and objects, include harmonisation between the designing object and the environment at all stages of its life cycle both in consumed resources and effects on the environment, as well as considering the need for further natural utilisation. One of the important factors to implement this principle in designing is a criterion of power consumption of an aircraft for the entire life cycle. This criterion was developed in the 1970s under the guidance of Professor V.G.Maslov. It corresponds to the current trend of energy saving. Its development will help to evaluate new objects from fundamental positions of use of available energy resources.

3. The Creator's plan: Designing principles.

We cannot understand the gods
Though the thoughts are pure…
Isaro

We cannot predict
How our word will respond
And sympathy is given
As we are given grace
F.Tyutchev

In the beginning was the Word,
And the Word was God.
From the Bible

We must be courageous enough and incredibly skilful at the same time to try to unravel the Creator's plan, or at least provide the hypothesis of his ideas, his plan. However, the atheistic upbringing allows the author not to be shy and remove from himself the taboo on this subject.

Probably, the Absolute's idea was to create a stable, self-developing, evolving World from its lowest levels, physical worlds, to the highest one, the level of Creator of the Universe. And the evolution of Mind from its inception to full self-awareness of ourselves in the world is the most important thing in all this. The newborn mind (our included) is an ignorant child who is sent by the Creator to a long trip of learning the World in order to experience its laws and learn to live in accordance with them. It is not our will to challenge the laws of the Creator, or try to correct them. He chooses life according to them. His choice is the basis of evolution.

The conditions for the evolutions of mind are created only when there is a free choice of two extreme opposites, in the presence of polarity. These opposites Man associates with the concepts of good and evil, creation and destruction, light and darkness. Polarity is of course a relative concept. The choice that the nature or so-called Creator, an ontological essence, makes can be as different and endless as the halftones in the spectrum of light. The conditions where a new entity is formed (geological, biological, or artefact), largely determine its properties.

The first aircrafts that looked like an extinct pterodactyl, seem to us somewhat clunky compared to today's airliners. Millions of years of natural evolution (selection), a long way of trial and error, are compressed into hundreds and tens of years, when the accumulated knowledge implements the artificial selection, which does not require experimental verification of options of evolution.

Evolution and ***natural selection*** are supplemented by ***scientific and technological revolution*** and ***artificial selection*** that Man mastered. In fact, absorbing the accumulating knowledge Man is able to speed up the selection, and in some cases even to participate in shaping the development path thinking of satisfying the ***needs***.

Man like nature can make mistakes. Moreover, different people having equal conditions, act differently. The notorious "human factor" defines our being.

DO NOT shoot the pianist,
He plays as he can.
Note in one of the bars in Texas

3.1. Design psychology

In designing there are always an object and a subject. The subject sets an objective, defines methods, means and selects resources to create a particular artefact, an object of designing. The personal characteristics such as preferences, predilection, perception, understanding, experience, form a need and estimation criteria of the future object of designing as well as define agents with their methods, subcontractors and their methods, consumers, and the market. Human factor plays a great role in the process which starts with an idea until the final costumer. “Cadres decide everything!” is not just a slogan of the time but an axiom of the evolution of human civilization.

Ontology of designing, its thesaurus and basic principles are invariant to the domain while the designing as an activity is always object-oriented. Ontology of designing summarises the experience of different domains.

It is believed, that in most cases when learning a discipline it is more effective to use as an example the field where the student has more knowledge, or at least a general idea about it. Therefore, the ontological analysis within the discipline is provided with the examples of designing an aircraft [6].

Design psychology is a part of applied psychology. Engineering psychology, that studies the processes and means of information exchange between a man and a machine, the psychology of management, cognitive psychology (and virtual psychology as its new subdiscipline), and acmeology are the closest fields to the psychology of designing. The last mentioned section of psychology is interpreted as the science of regularities and ways to achieve excellence in all types of individual human activity.

The main areas of engineering psychology (the closest to design psychology) are:

- Analysis of human tasks in management systems, distribution of functions between a man and automated systems;
- Research of the processes of receiving information by human;
- Analysis of the information processing by human, its storage and decision-making;
- Use the research results for the designing human-machine (human data) systems;
- Use the research results for virtual psychology.

As opposed to virtual and engineering psychology design psychology studies the process of formation and transformation of the needs into an imaginary virtual object. A need is always a stimulus for development.

“Nature does not like a vacuum”. Therefore, She or He (as we assumed the fact of His virtual existence) introduced a ***need*** into a biological model that is an inner state of psychological or functional sense of lack of something that appears based on situational factors.

Needs vary based on activity (needs in work, learning, leisure, etc.), objectives (material, spiritual, ethical, etc.), the subject of needs (individual, group, etc.) and so on. Usually a person has more than ten unsatisfied needs at the same time. The subconscious mind puts them in the order of importance forming a complex hierarchical structure aka Maslow's hierarchy of needs. Maslow divided the needs according to the order of satisfaction when the needs of the upper level appear after the largest and most fundamental levels are satisfied.

Physiological,
Safety,
Love\Belonging,
Respect,
Cognition,
Aesthetic,
Self-actualisation.

Objectification of needs is the process of "recognition" the subject or the modelling of it by a need. In this process the need becomes definite in this particular subject. Objectification is an important process where the motive that pushes us to act is born. The need gets its specification through objectification. Therefore, the motive is defined as an objectified need. When there is a motive, and the need is objectified the behaviours are dramatically changed. They have a direction based on the motive.

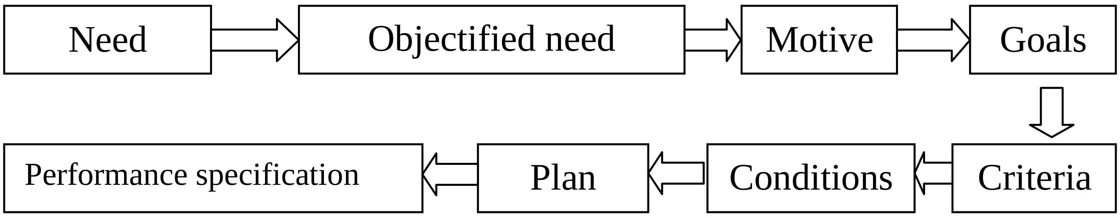


Fig. 3.1. Transformation of the need into Performance specification

The need is a driving force in the design process. It is closely connected with a man (customer, designer), a subject of designing. It is hard to predict how it will be formed but almost always it is based on the past: an experience or new associations based on stored information about the past, or the random combination of everything. Design psychology as part of the ontology of designing studies the relationship between the subject and the object of designing, the motives of relationship between the applied methods, and the process of transformation of the need into a project through the performance specification (fig. 3.1.).

A need is a good stimulus
An axiom of Borgest

3.2. Principles of designing

Historically, designing was used in the sphere of "creation" (house-building, shipbuilding, machinery, urban development, etc.) as the process related to making drawings, calculations on papers, computers, etc. of *the structure* and *functioning* of a future product (houses, ships, cars). With the development of "production" sphere the semiotic and intellectual activity based on drawings and calculations become more complex. It gained new functions like organisation of production, submission of plans and parts of manufactured products, linking the requirements with the products (on the drawings), evaluation and selection of the best decisions, and others. At this stage all these functions were formed within the "production" and were not perceived as independent.

When the division of labour between the architect (designer, engineer, draftsman) and the manufacturer (builder, mechanical engineer) takes place designing becomes an independent field. The first becomes responsible for semiotic and intellectual part of the work (ideas, drawings, calculations) while the second takes the creation (according to the drawings). This designing activity is characterised by a number of aspects [36].

1. Fundamental division of labour between designing and production. The designer must develop a product considering the look and structure of the object as well as production process, linking the requirements with the product. The maker creates the product in accordance with the project.
2. The designer develops the product in semiotic schema using drawings, calculations and other symbolic tools (models, diagrams, photos, etc.). A reference to the object (the prototype or developing object) can be incidental and indirect.
3. Designing is characterised by a specific "logic" and special abilities. A designer can combine and apply the opposite or divergent requirements for an object, develop plans and subsystems of an object, describe the form, functions, operation and structure of an object independently of each other and then combine them, design different versions of an object and its subsystems, compare these options. When developing a product designers create a kind of "semiotic models". The models of the designing object from previous stages are used as tools in creating the models of subsequent stages of designing.

Designing and science are opposed and identified at the same time. For example, P.Hill writes: "Engineering design can be considered as a science. Science is generally understood as generalised and systematic knowledge." However, designing is fundamentally different to science and engineering. First of all, they differ in objects. The object of scientific research is knowledge, and a project is the product of designing. Secondly, there is a distinction in methods of creating the product. Designing includes knowledge created by science, and science has a number of elements from designing (designing of mental and technical experiments, their equipment, etc.). However, the fundamental difference is still there.

The project organises the manufacturing activity, and knowledge satisfies the cognitive needs explaining the unknown (new) content by the known. The efficient creation of new models (mathematics is one of the main sources of efficiency) followed by proof of the effectiveness of the model relative to the object is a characteristic of obtaining scientific knowledge.

As opposed to science, designing does not have cognitive purposes. The aim of designing is to create an object that meets certain requirements and possesses specific qualities. It is designed based on symbols and knowledge. Knowledge is just a means for the designing. With its help (based on descriptions of the prototypes, functions, structures, relations, norms, etc.) a designer composes instructions to create an object (*a project as a system of regulations*) and describes its structure, function, internal and external views ensuring that it meets the requirements of the customer and the principles of designing (*a project as the model of a developing object*). The project as a model has two main functions: "communicative", that connects the customer, the designer and the consumer, and "object-ontological" which ensures the creation of the designing object within the process of design. [36]

Design drawings as a complex symbolic device has the ability to express two different groups of meaning and content at the same time. Due to this the project can be read the first time as "knowledge and description" (client – designer – consumer), and the second time as a complex instruction (in the production activity). The drawing can be divided into elements, parts, fragments, between which different relationships are set such as equality, similarity, part-whole, proportionality, etc.

The capability of not referring to the developing object, or to the test of its properties and characteristics in practice is one of the conditions of designing effectiveness. This fundamental feature of designing is achieved by knowledge (scientific, engineering, or experimental). The knowledge already has the basic functions and structures of designing and the relations connecting the functions with the structures. Normally, designing is a process from the requirements to the functions (functioning) as well as from the functions to the structures. During the designing some functions are split into others and complex structures are isolated from simple ones and, vice versa (the stage of design analysis and synthesis).

An engineer determines the types, characteristics of the functioning and structure of the object as well as relations between functions and structures. It means that he or she gets the knowledge which a designer puts into the base of operations of analysis and synthesis, specification, and the development of the project solutions as well as its estimation. If the engineering works are not developed yet or "delayed", a designer turns to experts (manufacturers, plant operators) to get experienced knowledge for designing.

The generalisation and analysis of the experience in designing as well as some researches are directed to obtain experimental knowledge. For example, the calculation of strength, stability (in aircraft designing), resistance and voltage (in electrical engineering) is based on advanced engineering and technical sciences, while the “calculation” of traffic flows, people’s behaviours and functioning in “man-machine” systems is based on experimental knowledge and ideas (descriptions of prototypes, observations, hypothesis, etc.).

Designing crowns the long evolution of technology and engineering. For the first time it connects the development of semiotic models (scientific knowledge and theories) with action, forming a united process of engineering. In this field, again for the first time, the procedure of the direct satisfaction of requirements for future products is observed. However, an engineer is concerned and limited by connection in the product between two principles, *natural* and *technical*. The first one is a source of energy while the technical principal is the ability to bring these natural processes to life.

As opposed to technology and engineering, designing does not refer to a real material, a product, or experience. When the production is organised through projects it is also released from technical steps. Designing is mainly an art. Here, a product is created with models and instructions from the beginning to the end. The ability not to refer to the materials, product, and experience as well as to decide the product based on operations with symbols and on models and to compare the possible solutions allows shortening the production time and increasing the quality of the product. Specifically in designing, different requirements for the product are quickly and effectively satisfied. Actually designing is the first and the basic mechanism in modern culture, that connects production with consumption and a client with a manufacturer.

There are more principles of designing in addition to temporal and ecological principles [35, 36].

The principle of independence: the project implementation does not modify the nature and its laws (but changes the environment).

The principle of feasibility: having a project, a product (a thing, an aircraft, a construction, a city, a system, etc.) can be created at the current production.

Correspondence principle: the functioning processes and morphological units (building units) can be identified, described and compared in a designing object. The same applies to the functions and constructions.

The principle of completeness: even though almost all the projects can be improved and optimised, however, it meets the basic requirements from the customer and society in general.

The principle of structural integrity: a designing object consists of elements, units and relationships that can be made in existing production.

The principle of optimality: a designer seeks to optimal solutions.

Typological principle: designing is an object-oriented activity. [35]

The well-known architectural principle “function is followed by construction”, or “function is followed by form” is true for classical mechanics. However, in electrical, information and other fields it is not applied; there are other principles of project life organisation. Each field has its own principles of organisation and functionalism. [35]

The concept of project culture was based on the assumption that there is an exchange of schemes, concepts and procedures between different types of directionality that are connected by the project activity. Design methodology was developed in a particular subject field. Therefore, the concept of project culture was introduced in order to promote the general methodological understanding of designing. For this purpose, it was necessary to describe the phenomenon of exchange of concepts between different fields of designing and integrate them together. Ergonomic design is the leader in project culture. Then video culture developed; and now there is IT-technology.

There are different forms of directionality of subject:

1. Methodological scheme describes how the activity in general and, in particular, project activity are organised;
2. The specific principles of formation, principles of organisation of the project activity.

Ecological principle is one of the methodological principles of designing. Designing is always an environmentally related activity. A situation is singled out according to specific features, which usually set performance specifications for designing. It is done in order to introduce something in this specific environment in the way that the environment could retain its own properties and the introduced “object” could somehow operate within it. Later, in the process of mutual adaptation, they can affect each other. Feasibility and implementation as well as the object itself are the main constitutive features on the phase of understanding the directionality. As the implementation means the ability of ***replacement***, the complex systems can be designed in parts that can be replaced using the block method. This modularity and the ability of replacement provide manufacturability. Therefore, designing is oriented on replacement thereby on modularity, thereby on manufacturability. Logistics is the aim of technologizing of designing. Ontology is brought to logistics, to the list of entities which motion in time and space is methodologically recorded and controlled. [35]

Engineering designing developed on the basis of natural sciences and mathematics. It became a separate object-oriented activity through object-ontological form, as an object is designed in ontological images. Directionality singles out a situation in an environment.

Directionality is inherent in culture and the environment where we live as well as in designing as its result or functionality. ***The anthropic principle*** states that the universe is organised in the way that if it was designed suitable for Man. This cosmological principle is not related with the creation of objects, but it shows the development of project culture, and the ontological meaning of directionality. [35]

The underlying objective of designing along with all project communications and reflections is the maintenance of sufficient ***project readiness*** as well as the readiness to development and processing. Therefore, a strategic reason of designing any projects is to develop different schemes of project activity to the state of readiness. Here, the designing links up with strategic thinking, strategic planning; different types of directionality are combined together and the reflexive critical understanding of the readiness is comprehended. [35]

The past is no longer, the future is
not yet, and the present is fleeting.
Aurelius Augustinus

3.3. Designing is the future planning

3.3.1. Human potential

Innovative potential in some degree is characteristic of almost every person; however, almost in all the cases it is defeated by usual stereotypes. Almost always those who have the innovative power, in the reason of the nature of innovation, are just a few people, can be even one single person. The number of people that are capable to do business, i.e. to create, produce, bear responsibility, is one out of 10. The proportion of people that have aptitude towards science are even less. Lucky and talented people can be counted on the fingers on one hand.

Throughout human history, from ancient times to the present day, just one out of thousands of ideas that promise different benefits if it is implemented, is really constructive. And even when it is realised the result could be completely different to the initial plan of the proponent. This is the way of progress. As for other ideas they turn out to be unrealistic, or even socially dangerous and harmful.

While conventional thoughts are common for “normal individuals” of similar population, innovative ideas are related to “mutants”. The more original idea the more disgusting the “mutant” is for normal individuals. Mutation (that is optimal for changed conditions) is the engine of progress. Human would not derive from apes or anyone else without mutation.

The idea that generates innovation and is positive and constructive should pass the “test” to be viable. The concept of scientific and technological revolution fundamentally changes the human life.

3.3.2. Futuraphobia effect

The human “fear” of radical decisions and conclusions has influence on social prediction and project planning. This actually is the “futuraphobia effect”. It means that the human psyche reacts extremely irritating to any “picture of the future” that is significantly different from the present. As a rule this picture causes a negative attitude, therefore, the future appears as a slightly idealised present. It is important that “futuraphobia” as rejection by human of any image about a different future that is different from the usual reality, has a negative impact on social prediction and project planning.

It would be appropriate to mention the famous “London forecast” about horse drawn carriage development in 50 years made in the 1880s. According to it, by 1930 the capital of England would have been covered with a two-meter layer of horse manure. The prediction failed. This was due to incorrect use of long-term linear extrapolation of existing data, and “fear” to assume an “invention” of many technological innovations, that dramatically changed the course of events and led to the development of new kinds of transport.

In addition, probably there was a lack of awareness among scientists about the transport sector and the trends of its development that negatively affected the situation. [34] And what about the statements of Lord Kelvin made in 1896 about the impossibility of aeronautics except using hot-air balloons, or the statements of air force minister in 1935 that jet propulsion is not competitive with the pair engine-propeller? Rephrasing the famous expression, the forecast is a delicate matter!

3.3.3. The Oedipus Effect

The well-known Sophocles' tragedy about Oedipus the King that starts with the words: "Not the man's freedom to do what he wants, but to take responsibility even for what is not wanted". It tells about the development of events that were given a particular prediction. Literally, it talks about the effect which refers to the prediction of the development. It shows that the knowledge about the future changes the future itself. Ideas are materialised!

Designing is the key form of specification on prediction. Its features and advantages comparing to planning and programming are the combination of diagnostic and standard approach, the absence of clear dominance of the "must" image over the diagnosis of the situation and a realistic assessment of available resources, and unnecessary details of objectives, results and work methods. In other words, a design decision is not a regulatory document. At the same time it develops a model of "proper" future in accordance with given constraints on time, resources and the quality outcome results. It correlates the problem with a general way to solve it allowing for multi-variant in achieving the goals as well as sets reasonable time frames in problem solving depending on characteristics of a problem situation. [34]

In fact, designing is a technology that represents a constructive, creative activity that identifies problems, sets goals and specifies the procedures. This technology and the practical implementation of the project are two opposing "flows" of intellectual activity. In real situation the process of the finalising the project decision requires the subject of designing to mentally reproduce a "direct" algorithm of design activity as well as a "reverse" sequence of developing system. This allows to "design" the final result, to calculate the effectiveness of problem-solving techniques taking into account available (or expected) resources. [34]

It should be noted that not just the constructive phase of the project development but goal setting as well is a kind of creative process. The objective of the project ("desired" image) is always beyond the frames of real conditions of subject-object domain. Based on real social problems, conflicts and resources, it includes the potential abilities to make some social changes.

3.3.4. The intuition phenomenon

Intuition in prediction is the reality of human consciousness. Usually intuition is defined as "the highest revelation", an unconscious apprehension of the future with the help of the subconscious. It can be irrational, based on feelings, or can have rational basis of human mental activity ("intellectual" intuition). Some examples of "intellectual" intuition, insight, can be associated with various prophecies that can be found in different works of philosophers, the pillars of the world science fiction such as Jules Verne, I.A. Efremov, as well as famous scientists like Archimedes, Isaac Newton, Jules Poincare, and D.I. Mendeleyev.

However, intuition as a method of reality cognition has some limitations. If it is effective in respect of one situation it is transferred to the others without a proper understanding and it may turn to prejudice. It also can turn into a project-mongering if it is not supported by significant data base, or can degenerate into a kind of charlatanism that is based on speculation and freely interprets the events.

4. Ontology as a specification of a conceptualisation

The concept of “ontology” has two meanings. The essence of traditional ontology:

- A philosophical discipline that studies the most common characteristics of being and entities;
- An artefact, a structure that describes the relationship among the elements of a system.

Ontology is a system that consists of a set of concepts and a set of statements about these concepts based on which classes, relations, functions, and individuals can be described.

One of the most famous definitions of ontology was provided by Tom Gruber: ontology is a specification of a conceptualisation which is, in turn, the structure of reality considered independently from the domain vocabulary and a specific situation.

A more recent definition by Gruber: ontology is a *formal specification* of a ***shared conceptualisation***. It means that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group.

4.1. The components of ontology

The basic components of ontology are:

- Classes (or concepts);
- Relations (or properties, attributes);
- Functions;
- Axioms;
- Individuals (instances).

Classes, or ***concepts*** are used in a broad sense. Any entity that may be given any information can be a concept. Classes are abstract groups, sets, or collections of objects. Classes may classify individuals, other classes, or a combination of both. Usually in ontology, classes are organised in ***a taxonomy***, that is a particular classification arranged in a hierarchical structure. For example, plane and helicopter are subclasses of aircraft which, in turn, is a subclass of aviation complex or transport.

Relationships (also known as ***relations***) between objects in an ontology specify how objects are related to other objects. An example of a binary relation is a “part-whole relation”. Such units as a wing, a fuselage, chassis are parts of the whole, a plane.

Functions: complex structures formed from certain relations that can be used in place of an individual term in a statement. A thrust-to-weight ratio can be an example of functional relationship, which is calculated according to length of take-off, wing loading, and lift coefficient.

Axioms are used to record the statements that are always true. They may be included in the ontology for different purposes such as to determine the complex restrictions on attribute values and the arguments of relations, to verify the information described in the ontology, or to display new information. The well-known mass balance equation is an example of axiom: the sum of the relative weights of aircraft parts is equal to one.

4.2. Reasons for creating ontologies

In recent years the development of ontologies has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have become common on the World-Wide Web. The ontologies on the Web range from large taxonomies categorizing Web sites to categorisations of products for sale and their features. Many disciplines now develop standardised ontologies that domain experts can use to share and annotate information in their fields.

An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them.

- To share common understanding of the structure of information among people or software agents;
- To enable reuse of domain knowledge;
- To make domain assumptions explicit;
- To separate domain knowledge from the operational knowledge;
- To analyse domain knowledge.

Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies. For example, suppose several different Web sites contain transport information or provide transport e-commerce services. If these Web sites share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sites. The agents can use this aggregated information to answer user queries or as input data to other applications.

For example, in the aircraft pre-designing, the published values of such aircraft parameters as take-off weight, gross weight, or empty weight of the same aircraft depend on specific parameters such as distance, aircraft loading, fuel, etc. When developing ontologies it is important to take into account the interrelationship and interdependence of these characteristics.

Specifications are also needed for the values of efficiency of engine components while designing CCD. For example, the value of efficiency of polytropic or adiabatic compressors used in a data base or by a designer as well as the working conditions significantly influence on the result.

Therefore, the main task of an ontology is the exact specification of the domain.

Enabling reuse of domain knowledge is one of the driving forces to study ontology. For example, if one group of researchers develops such an ontology in detail, others can simply reuse it for their domains. Additionally, if we need to build a large ontology, we can integrate several existing ontologies describing portions of the large domain.

Making explicit domain assumptions underlying an implementation makes it possible to change these assumptions easily if our knowledge about the domain changes. Hard-coding assumptions about the world in programming-language code make these assumptions not only hard to find and understand but also hard to change, in particular for someone without programming expertise. In addition, explicit specifications of domain knowledge are useful for new users (these include undergraduate and postgraduate students) who must learn what the terms of the domain mean.

Separating the domain knowledge from the operational knowledge is another common use of ontologies. Analysing domain knowledge is possible once a declarative specification of the terms is available. Formal analysis of terms is extremely valuable when both attempting to reuse existing ontologies and extending them.

4.3. Ontology classification

The ontology of designing can be roughly divided into two areas:

- Ontology as a formal system based on mathematically prices axioms;
- Ontology as a system of abstract concepts that exist only in human mind and can be expressed in natural language (or using other system of symbols). This direction developed in computational linguistics and cognitive science.

Hence, there are two alternative approaches to creating and researching ontologies. The first (formal) approach is based on logic (first-order predicate, descriptive, modal). The second one (linguistic) is based on the study of natural language (semantic in particular) and on the ontology development in large text arrays.

Nowadays, these approaches are in close relations with each other. The connections that allow combining the appropriate methods is searched. Therefore, it is sometimes difficult to identify lexical ontology with elements of formal axiomatics and logic systems with linguistic knowledge.

Regardless of different approaches the classification of ontology can be divided into three main principles:

- the degree of formality;
- the purpose of development;
- the content.

4.3.1. Classification by degree of formality

This classification can be presented in the context of “human-understandable” and “machine-understandable” descriptions. On the degree of formal presentation and the use of certain syntactic structures some terms can be stated. Such as:

Controlled vocabulary is the final list of terms (e.g. subject indexing scheme). Catalogues provide a precise (one-valued) interpretation of terms. For example, each time referring to the term “apparatus” we will use the same value, regardless of whether it is in the context of “flying apparatus”, “Ilizarov apparatus”, or “the apparatus of government”.

Glossary is a list of terms and their meanings which are presented as comments in natural language. This provides more information and helps to better understand the term. The terms' interpretations can be multivalued. Glossaries are not suitable for automated processing by software agents. However, it is possible as in a controlled vocabulary to assign an ID for a term.

Thesauri provide some additional semantics in their relations between terms. They provide information such as synonym relationships. Early web specifications of term hierarchies provide a basic notion of generalisation and specialisation. Yahoo, for example, introduced a small number of top-level categories such as "apparel", and then the category "dresses" as a kind of (women's) apparel. The explicit hierarchy of Yahoo did not match exactly the formal properties of the hierarchical relationship "subclass-class". In these hierarchies a situation in which an instance of a descendant is not an instance of an ancestor may occur. For example, the general category "apparel" includes a subcategory "women" (which should more accurately be titled "women's apparel") which then includes subcategories "accessories" and "dresses". It is clear that the accessories such as "badges" are not instances of dresses. ***Transitivity*** (the transition of properties, such as, the equation a=b is transitive. Since a=b and b=c, then a=c), an important property of the relation "subclass – class" is not true in this case. Similarly, in aviation, when the general category "aircraft" has a subclass "civil aircrafts" that, in turn, has one of the subclasses "payload". As in the previous example, "aviation bombs" as a payload but for "military aircrafts" is not related to "civil aircrafts".

Formal taxonomy is a kind of ontologies that has a precise definition of the relationship "subclass-class" (or ISA). In such systems transitivity of the ISA relations is observed. In these systems if A is a subclass of B, then each subclass of A is a subclass of B. Strict subclass hierarchies are necessary for exploitation of inheritance.

The next point on the ontology spectrum includes ***formal instance*** relationships "instance-class" (IsInstanceOf). Some classification schemes only include class names while others include ground individual content. For the relationship "instance-class", if A is a subclass of B, then if an object is an instance of A it necessarily follows that the object is an instance of B. So, in the given above example "badges" cannot be placed below "apparel" in the hierarchy, or became an instance of this class. The same is about "payload". It is a general class that isapplied to "civil", "military", and other purposes of "aircraft". In this case, "payload" itself defines the function of "aircraft", therefore such instances of the class as "bombs", and "troopers" should not be a subclass of "civil".

A ***slot*** is an element of a class (a frame). Classes may have information about the properties (slots). For example, "aircraft" may have such characteristics as "price" and "made in". The properties are even more useful when they are defined on the upper levels of the hierarchy and inherited by subclasses. Thus, in the consumer hierarchy the class "product" may have "price" which is received by all the subclasses.

Designed ontologies often have some restrictions on slots. The values of properties are taken from a predefined range (integers, constants), or from a subset of ontology concepts (the set of instances of the class, a set of classes).

When describing more complex facts, the ontology and its structure get more complicated. For example, it may be required to define properties of an instance using a mathematical expression that is based on the properties of this instance and the properties of other instances. In some cases classes do not have common instances. It means they are disjunctive.

4.3.2. Classification by goal

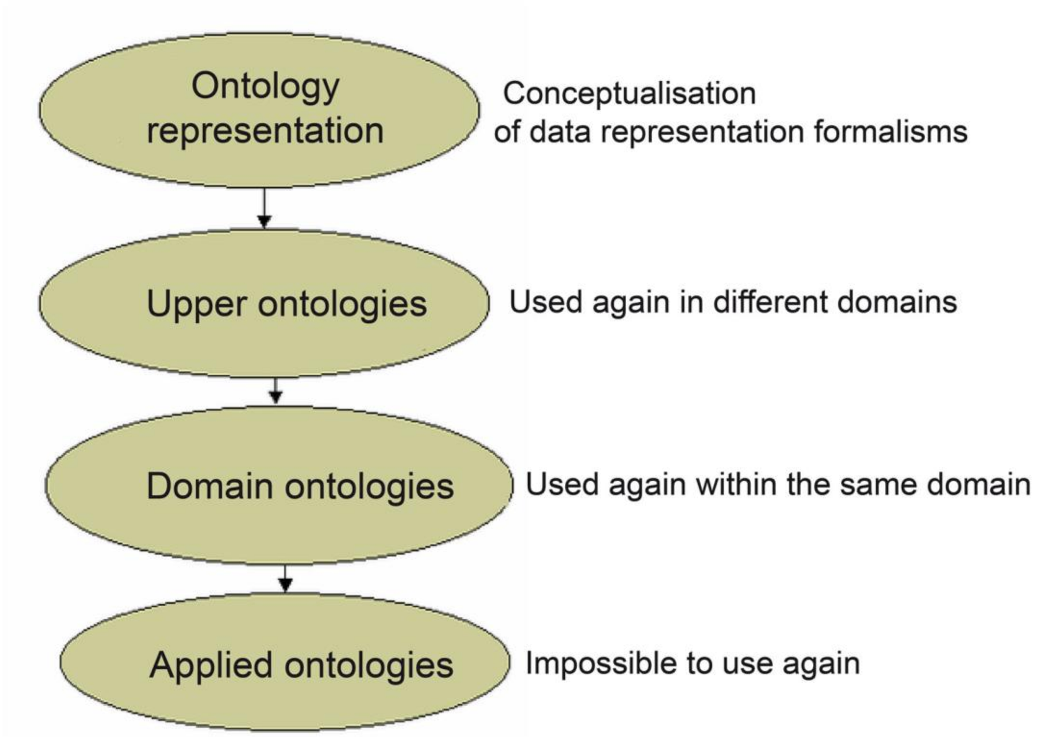


Fig. 4.1. Ontology classification by goal [26]

4.3.2.1. Ontology representation

It aims to describe the area of knowledge representation and to create a language to specify the lower-level ontologies. Example: the description of the OWL concepts with RDF/RDFS (fig. 4.2.). In this description such concepts as “class”, “relations”, “restriction”, “data range”, etc.

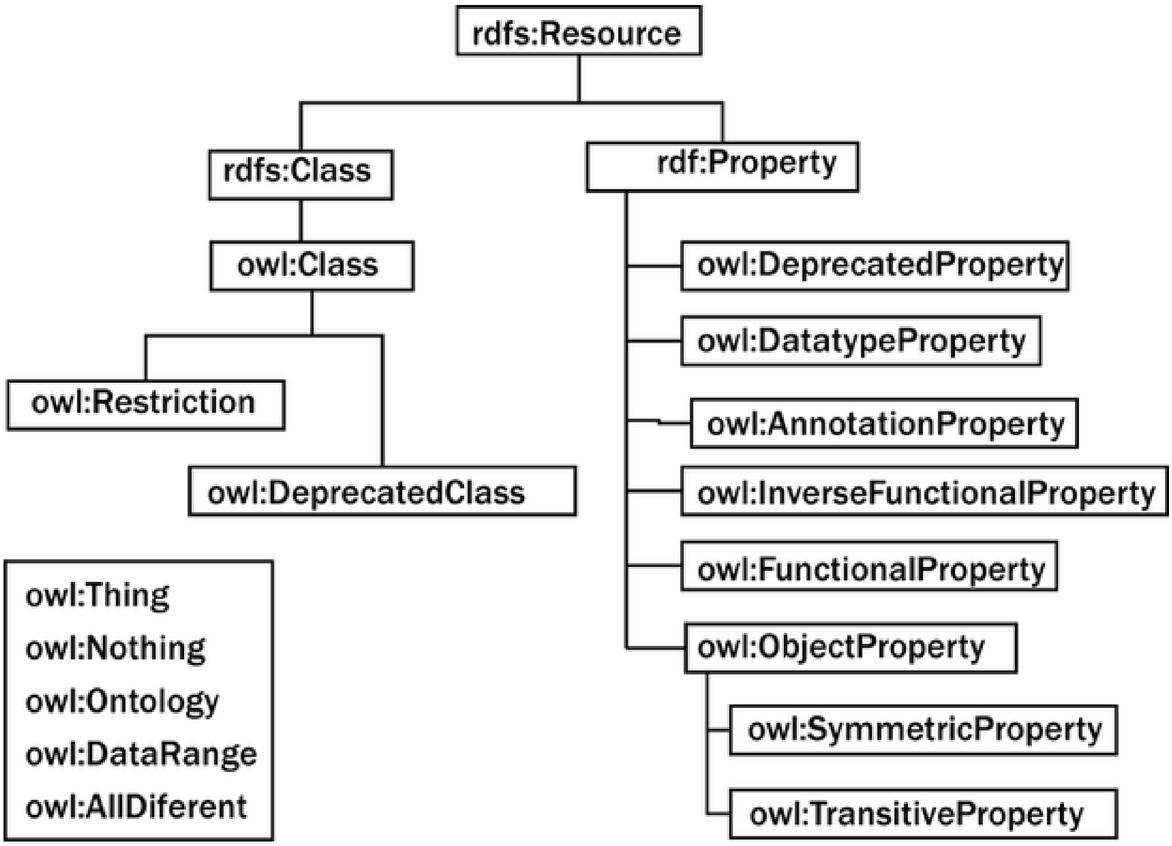


Fig. 4.2. Ontology representation using OWL [26]

4.3.2.2. Upper ontology

It aims to create for multiple uses a general ontology that describes general concepts, the same for all knowledge domains. The attempts to make such an ontology were not successful so far. Many upper ontologies are similar to each other; they contain the same concepts like entity, phenomenon, process, object, etc. There are several large upper ontologies: Cyc, DOLCE, SUMO (fig. 4.3.), J.Sowa's ontology and others.

4.3.2.3. Domain ontology

It has similar purposes to upper ontology, but it is simile by just one domain, for example, aviation, medicine, culture, distance-learning, Internet-based technologies. Domain ontology generalises the concepts that are used in the domain but is not concentrated on the task itself (for example, the ontology of an aircraft is independent from any specifics of particular plane models). Standard ontologies that can be used by domain experts are being developed in many disciplines.

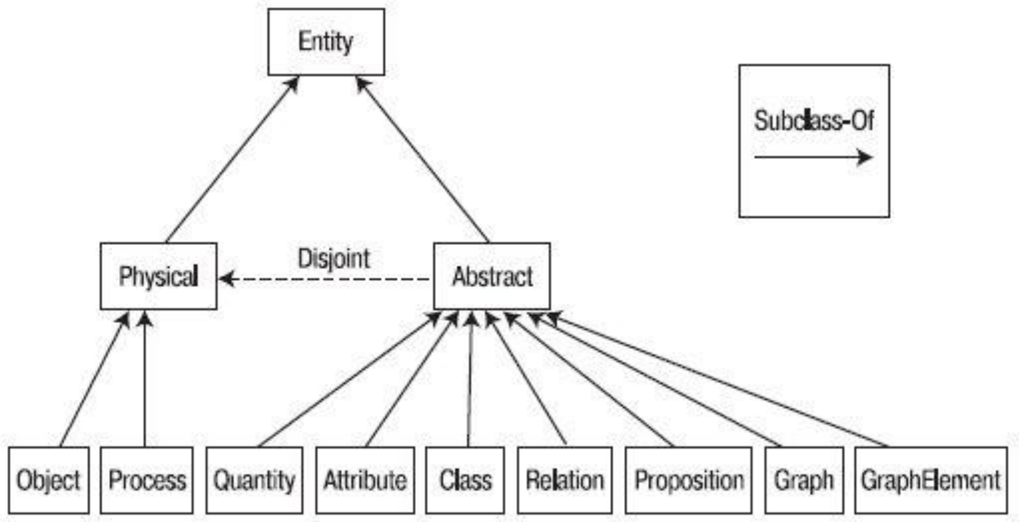


Fig. 4.3. Upper level of the SUMO ontology

4.3.2.4. Applied ontology

It aims to describe a conceptual model of a specific task or application. Applied ontologies describe concepts that depend on the ontology of the tasks and the domain ontology. For example, ontologies of domain elements such as airplanes, cars, building materials, computers have specific data (e.g. TOVE, Plinius).

The goal of the TOVE (Toronto Virtual Enterprise) project is fourfold:

- Create a shared representation of the enterprise that each agent in the distributed enterprise can jointly understand and use;
- Define the meaning of each description (semantics);
- Implement the semantics in a set of axioms that will enable TOVE to automatically deduce the answer to many “common sense” questions about the enterprise;
- Define a symbology for depicting a concept in a graphical concept.

The model is multi-level spanning conceptual, generic and application layers. The generic and application layers are also stratified and composed of micro theories spanning, for example, activities, time, resources, constraints, etc. at the generic level.

The goal of the Plinus project is semi-automatic knowledge retrieval from the text in natural language, in particular, from the literature about the mechanical properties of ceramic materials. As the texts have a broad range of concepts, many generic ontologies that cover such concepts as ceramic materials and their properties, methods of processing, various defects of materials, etc., are required.

4.3.3. Classification by content

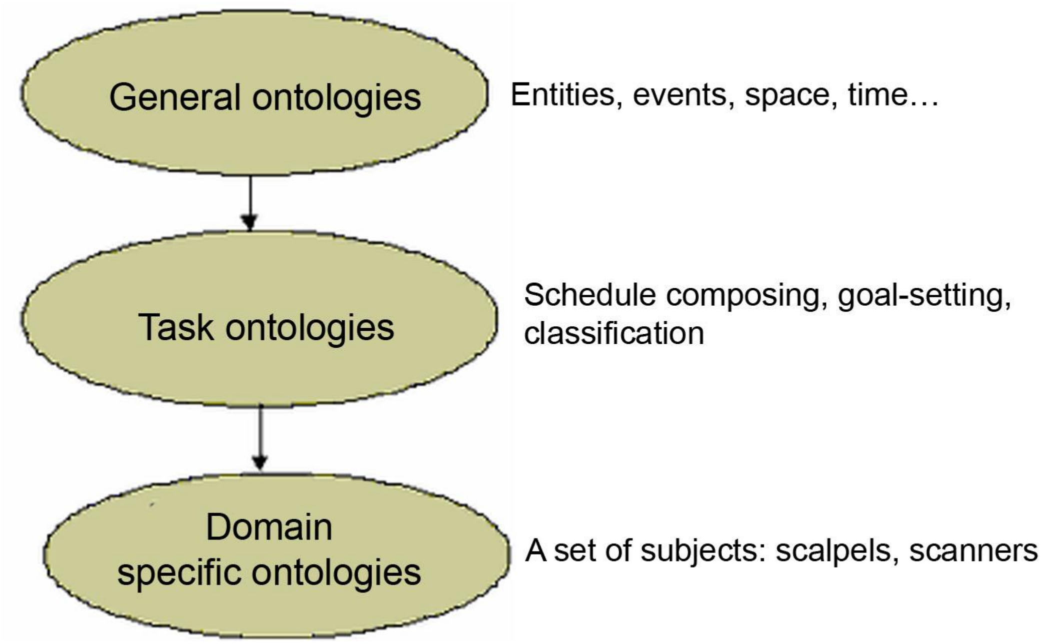


Fig. 4.4. Ontology classification by content [26]

General ontologies describe the general concepts (space, time, object, event, etc.) that are independent from a particular problem or domain. Representation ontology and upper ontologies belong to this category.

Task-oriented ontology is used by a particular applied programme that contains terms, which are used to develop software with a particular task. It shows the features of the application, and also it may contain some general terms (for example, in a photo editor there are special terms such as graphic palette, fill type, blending, etc. as well as general terms as save and load a file). The tasks may be very diverse: schedule composing, goal-setting, diagnostics, sales, software development. Task ontology uses the specific of upper ontology terms (general ontologies).

Domain specific ontology describes real objects that are involved in any activity (production). For example, it could be an ontology of parts and components of a particular aircraft producer (Tu, or IL) and the information about their suppliers, characteristics, etc.

5. Ontology in use

There are some advantages of problem-solving ontology:

- In calculation (reduction of computing time);
- In economic terms (cost reduction in software development, data integration).

The best proof of the benefits of ontological approach is to compare two functional projects. However, such a task is beyond the abilities of any single organisation. Information retrieval is an important task.

In modern search engines the texts are automatically indexed by a set of words that make up these texts. Such a text representation as bag of words has a large number of obvious cons that make the search of relevant texts difficult. For example,

- *Redundancy*: in the word by word index synonyms that express the same meanings are used;
- Words *independence* from each other in a text that does not match the properties of the linking text;
- *Multiplicity* of words: as ambiguous words may be regarded as a disjunction of two or more concepts that express different meanings of the ambiguous word, it is unlikely that all the elements of the disjunction are of user's interest.

Concept indexing does not have these disadvantages. Here, the text is indexed not by words but concepts which are discussed in the text. With this technology:

- All the synonyms are narrowed to the same concept;
- Ambiguous words are referred to different concepts;
- The relations between concepts and corresponding words are described and may be used in the text analysis.

In order to implement the scheme of automatic concept indexing and concept search, it is required to have a means describing the system of the domain concepts, i.e. **domain ontology**.

The use of ontologies for information retrieval in broad domains has several features:

- The ontology should have a larger dimension;
- The concepts of ontology should have established links with linguistic units, i.e. the terms of domain;
- The methods of information retrieval based on ontologies should correspond to the methods of informational retrieval based on word by word methods in the whole search engine;
- The information retrieval task involves the use of ontologies for the analysis of linking texts, for which there are no well-developed methods of automated processing.

All these factors limit the implementation of ontology in search engines. This means that new researches are required:

- To find out what kind of ontologies can be more effective for information retrieval tasks;
- To develop information retrieval technologies that combine knowledge based methods described in ontologies and word by word methods;

- To develop methods of “rapid” creation of broad domain ontologies based on text collections and knowledge integration of existing ontologies.

These difficulties are also typical in the field of integration of heterogeneous database. Here, the benefits from solution introducing based on ontologies may be more obvious for a particular enterprise, a particular domain. As the system of concepts is limited by a domain and data volume is not as great as in Internet, it is possible to create a relevant ontology, because it is not important for an application (or a database) which upper ontology in particular they share with others. However, it is important to share one upper ontology or even one domain ontology that encourages database integration, ensures *interoperability*. The last means the ability of a system or a product to work with other systems or products without special effort on the part of the customer.

Consider some fields where ontology is implemented.

5.1. Semantic Web

The term was coined in 2001 by Tim Berners-Lee, the inventor of the World Wide Web, which oversees the development of proposed Semantic Web standards. He defines the Semantic Web as "a web of data that can be processed directly and indirectly by machines." In order to communicate with each other these agents must have a shared formal data representation for any resource. Ontologies are used to represent general, explicit and formal specification of values in Semantic Web.

A series of standards and recommendations has been developed as well as many projects have been implemented. However, despite some progress, it cannot be stated that the idea of Semantic Web was realised in practice.

In 1997 the W3C has defined the specification of RDF (Resource Description Framework). RDF is a simple language for expressing data models usually in triple-based structures (subject, predicate, object) and URI. In 1999 RDF acquired the Recommendation status. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web. RDF provides a minimum level for knowledge representation in the Web. RDF specification is based on earlier Web standards:

- Unicode represents the symbols of alphabets of different languages;
- URI is used to identify unique resources;
- XML and XML Schema make sure that there is a common syntax used in the semantic web. XML is a general purpose mark-up language for documents containing structured information. All data in the semantic web use RDF as the primary representation language. The normative syntax for serializing RDF is XML in the RDF/XML form.

RDFSchema (RDFS) was developed to describe structured vocabularies for RDF. It provides a minimum set for ontology specification. In 2004 it got W3C’s Recommendation. However, there are just a few documents written in RDF/RDFS.

In 2004 a new language, GRDDL, was formed. The GRDDL (Gleaning Resource Descriptions from Dialects of Language) mechanism allows existing material (including microformats) to be automatically interpreted as RDF, so publishers only need to use a single format, such as HTML. Software for Semantic Web was developed. As an example, Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. Another example is the SIMILE project. Focused on developing tools to increase the interoperability of disparate digital collections, much of SIMILE's technical focus is oriented towards Semantic Web technology and standards such as Resource Description Framework (RDF). In visual editing a large number of ontology editors began to support RDF.

In 2004 **OWL Web Ontology Language** became a formal W3C recommendation. An OWL ontology is an RDF graph, which is in turn a set of RDF triples. As with any RDF graph, an OWL ontology graph can be written in many different syntactic forms. The data described by an ontology in OWL is interpreted as a set of "individuals" and a set of "property assertions" which relate these individuals to each other. A large number of developed ontologies are encoded in OWL, and already existed ontologies are translated into it.

In 2005, the Rule Interchange Format (RIF) was developed. RIF is based on the observation that there are many "rules languages" in existence, and what is needed is to exchange rules between them (according to which a non-trivial logical conclusion may be done): Horn clause logic, higher-order logics, etc.

In January, 2008 SPARQL became a formal W3C recommendation. The Simple Protocol and RDF Query Language (SPARQL) is a SQL-like language for querying RDF data. It is widely used by designers of information systems.

A diagram, which is called Semantic Web Stack, or Semantic Web Layer Cake, is presented on the figure 5.1.

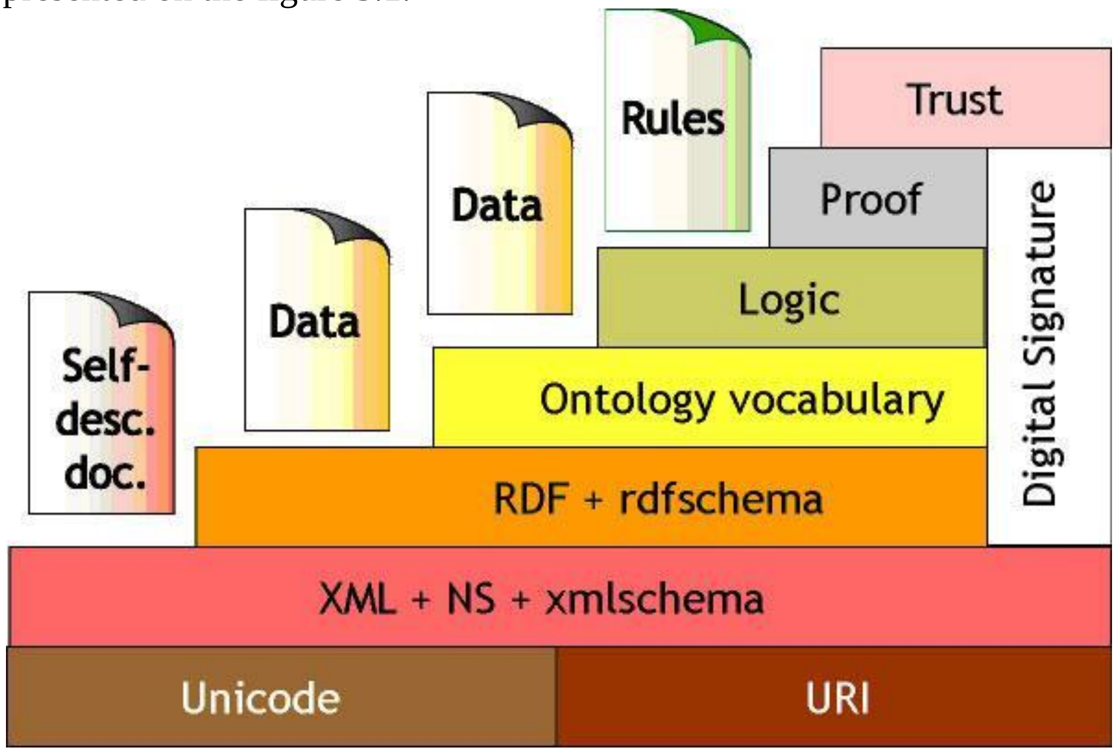


Fig. 5.1. Semantic Web Stack [26]

The Semantic Web Stack is an illustration of the hierarchy of languages, where each layer exploits and uses capabilities of the layers below. It shows how technologies that are standardised for Semantic Web are organised to make the Semantic Web possible. The layers “Ontology vocabulary” and “Logic” correspond to OWL and RIF. “Trust” at the moment is not adapted by any standards. Here one of the major obstacles to implement the idea occurs: maintenance of automated information check. Some providers of semantic descriptions may be tempted to “cheat” software agents providing false information, obtrusive advertising, etc.

Lack of working intelligent agents is another stumbling block for creating a Semantic Web. Not every programme in RDF is an agent of Semantic Web as well as not every programme written in Prolog is an Artificial Intellect application.

However, Semantic Web has been developing; new standards appear. The developing of RIF, setting the requirements and its area of use is a new step. The developing of new ontologies and integration of existing ones have been the main tasks for Semantic Web.

5.2. Information Retrieval

The basic concepts of document retrieval:

A **collection** is a set of documents that have any common properties (e.g. a collection of documents on a particular subject, or a collection of documents that have only a common format).

A document is a structural unit of information in terms of storage and retrieval from the collection. Text document can be represented as a sequence of smaller units such as paragraphs, sentences, words that are also documents in a particular context.

Information Retrieval (IR) is the area of study concerned with gathering, searching, retrieval and distribution of information with use of computer technologies.

There are some examples of Information Retrieval:

- Information Retrieval of documents queried by a user;
- Automated categorisation of documents by predetermined classifier;
- Automatic Document Clustering: documents of similar meanings are divided into clusters;
- Development of question-answering systems: the search of an exact piece but not the whole text that answers the question of a user;
- Automated document summaries, and many others.

The aim of IR is to satisfy a user’s needs in information. The information need itself is a psychological state of man. One of the problems of forming a query is that it can have several interpretations. It can be incomplete or redundant, has ambiguous words, depends on context, etc. Information need remains constant at any given time. It does not allow alternative or conflicting interpretations. Even though the query rarely corresponds exactly to the information needs, this is the only way of communication between a user and a search engine.

A search engine uses a query as input data to get a particular result, i.e. a sample from a collection of documents that matches the query (the search engine finds documents that are relevant to the query). This raises another problem: the user evaluates the search results according to the information needs but not the entered query. During the evaluation, it is decided whether the result is relevant to the user's information needs. Such an assessment can only be made by the user. The assessment on the relevance and the relevance itself are called true relevance. Relevance that is calculated by search engine based on its internal logic may not correspond to the true relevance.

There are other ways to determine relevance. Example: *thematic relevance* and *utilitarian relevance*. This document may accurately correspond to a user's information need on a particular subject (thematic relevance), but at the same time be useless for a particular activity within which a need occurred (utilitarian relevance).

From a user's perspective, the search engine starts the work when a query is sent. In fact, this is preceded by an important stage of indexing collections of documents. At this stage index tables are created, that significantly accelerates the query processing. The array data indexing is commonly used for faster access to information. An example of an index may be an alphabetical index or contents at the end of the book. Even bookmarks are a kind of index. Indexes are widely used to speed up the access to database. A feature of indexing in IR lies in the fact that the index that is required for full-text search in digital collections is more complete. It should contain all the terms that appear in the documents of the collection. The index, which has all the terms that appear in the documents of the collection, is called an *inverted file*. In many cases the inverted file contains just ***tokens*** (the roots of words) instead of all possible forms of each word. Numbers, alphanumeric codes, abbreviations are also considered as tokens.

The following characteristics are calculated for each token in the set of documents:

- The number of documents where this token appeared (it shows the prevalence of tokens in the collection);
- The frequency of occurrence of the token in the collection (it shows how "unusual" this token is comparing to others).
- In addition, the inverted dictionary stores the following information for the pair "token-document (that contains this token)":
- The frequency of occurrence of the token in the document;
- The displacement of the token from the beginning of the document. It can be measured in characters or words. This feature is often used for a faster retrieval of words from the context and creating a brief summary.

When retrieving documents relevant to the query it is necessary to determine how to process the query and calculate the relevance values for each pair "query-document".

5.2.1. Query processing

Historically, one of the first ways of query processing was a so-called Boolean searching. In this approach the words of a query are linked by logical connectives: AND, OR, NOT. Grouping with parentheses is allowed. Thus, the query is a logical formula, where terms or any additional conditions are atoms. A search engine based on Boolean search returns the documents where the formula (query) is true. Each atom of the formula matches the set of documents where the value of the atom is true. If an atom is a term, it matches the set of documents where this term occurs. Then elementary operations are done over the sets: union, intersection and complement.

This approach to query processing has several disadvantages:

- Search engine may return a lot of documents (or even all the documents of the collection). In this case a user should sequentially add conditions in the query to shrink the result set. The search is conducted by trial and error;
- Generally, a useful sample can appear when requesting a complex logical formula. In this case a user is required to know the rules of constructing formulae and the "language" of the domain;
- Due to the fact that there are only two values of relevance: true and false, the resulting sample cannot be organised by relevance. All the documents are equally relevant.
- All atoms of the formula have the same importance (weight), despite the fact that some are more important than others.

There are some ways to improve the quality of Boolean search. Thesaurus or another ontological resource can be used for automatic query expansion.

A negative side of the Boolean search is associated with the formalism of query processing. In order to neutralise them the approach should be modified. However, it does not mean that it should be given up. Many search engines use Boolean search as an alternative.

Ranked search is the main way to process a query by search engines in Internet. It is based on relevance calculation by means of query term frequency distribution over the documents of the collection. A query can be entered in natural language. During the pre-processing of the query particles and stop-words are deleted (for example, "where", "why", etc.). Terms are reduced to tokens. After that, based on the tokens a logical formula could be formed. However, experiments showed, that atom connection with the AND operation gives too few documents in the result set, and many relevant documents are remained outside ("spilling the baby with the water" situation). The connection of the formula atoms with the OR operation gives an opposite result: the sample is too big. In this case, the Boolean approach to the processing of natural language query is not adequate. Additional information on the relationship of terms (ontology) could be used to form a more complex logical formula.

It is offered to present every document of the collection as a vector in space that is equal to the number of tokens in the inverted file. The document is described by "weights" (coordinates) of the tokens. Axes are mutually orthogonal (tokens are pair wise independent and form a basis). The pre-processed query contains a sequence of tokens and, as well as any document can be expanded in the basis space. Next, to compute the relevance, a function that gives a relevant number of the interval [0,1] to each pair of vectors is determined. The extreme points correspond to “irrelevant” and “completely relevant” values. Intermediate values determine the degree of relevance of the query document (or two documents of the collection if it is required to find “similar” documents).

Another approach to query processing is based on a probabilistic model. It is an attempt to describe a ranked search in terms of the theory of probability. The problem is that the frequencies that are used in the ranked search are not related with the probabilities. The number of a term occurrence in a document cannot be a random variable and be used to estimate a probability of the term occurrence in other documents of the collection. Therefore, the frequency of terms occurrence cannot be applied for standard formulae of the theory of probability. This method of probability calculation is far from the true relevance and the useful relevance.

The model is based on the calculation of the relevance probability. If the probability is high enough, the document is considered relevant. However, their values may be useful when presenting the results (for sample sorting). Experiments showed that the quality of search engine work that is based on a probabilistic model is no better than those that are based on ranked search.

5.2.1. Evaluation of search results

With the increasing number of search engines, different techniques, and search algorithms, it became necessary to compare the quality of the results. Two terms were introduced: precision (p) and recall (r).

Precision is a share of relevant documents of the sample from all the documents in the sample.

Recall is a share of relevant documents in the sample from all the documents from the collection.

These two terms are usually conflicting. In practice it is impossible to have one hundred per cent of precision and recall.

Suppose N is the number of documents in the collection, n – the number of documents in the collection which are related to the query, m – the number of documents in the sample provided by the system according to the query, A – the number of relevant documents in the sample. Then, p=A/m, r=A/n.

	Relevant		**Irrelevant**
Extracted	A	B	A + B = m
Not extracted	C	D	C + D = N – m
	A + C = n	B + D = N – n	A + B + C + D= N

5.3. Integration of heterogeneous sources

Database (DB) is a collection of interrelated data that has some “hidden (internal) value”. Database is similar to knowledge base (KB) as they are also used to describe a certain domain in order to store, process and access information about it. However, there are some differences [1]. DB contains (and can process) large arrays of relatively simple information (the access is limited by the entered data). KB usually stores less information but has a more complex structure, which allows to get statements that were not entered. It is done due to inference.

Three operations can be applied to any KB: *define*, *tell* (i.e. to make a statement), and *ask*. Each operation can use one or more of their own languages, such as language for describing patterns and constraints, update language, query language, and language for describing the results. The advantages of using descriptive logic (DL) for the improvement of each of the languages are widely described in the knowledge base literature. To manage the data it is necessary to perform the following tasks:

- An expression of the conceptual domain model (ontology) for a particular data source;
- The integration of multiple sources;
- Query expression and processing.

First of all, it is necessary to describe the domain in the language that the description is well understood for a regular user and for designers. It is performed in a high-level language and has a requirements form. ER-diagram (ER-model) is the language for databases. In ER-model the world is represented as a set of entities (n-ary relations). The resulting *semantic model* can be stored on computer as well as the data itself. However, as a rule, it contains general and constant information as opposed to specific facts. This semantic model introduces the terms to describe the domain and determines their meanings by setting the relations and constraints. This way of data representation is close to ontology.

Logic scheme is created from a semantic model. It describes the data structure of database, data types, relations and constraints. Relational data model is the most popular and frequently used model to express logic scheme. Databases are used to specify logic schemes. In a relational model data is stored in tables that contain rows that, in turn, consist of cells with the simple data types values (integers, rows, dates, etc.). So, to describe a logic scheme it is required to describe the table names, their column names (attributes) and the corresponding data types. The relational database management system requires a set of columns (key) that uniquely identifies the row in the table to be defined. In the database it is not allowed to describe the disjunctive information. If there is no information about an attribute value, then it is considered equal to NULL.

Using the declarative query language SQL and defining the table content and structure is a universal and a powerful practical tool.

Database management system (DBMS) hides the details about function implementation from a user. It makes clear the physical level: the method of data storage on data carriers, auxiliary data structures to speed up access, and even the fact that database is distributed over several network nodes. However, a user may wish to obtain information from several independent (and probably heterogeneous) sources. In this case, a problem with linking various logic schemes into one may occur.

Integration of heterogeneous data sources is a fundamental problem that database developers have faced the last decades. The purpose of data integration is to provide one interface for various sources and allow users to focus on determining what they want to know. As a result, integration should free a user from searching relevant data sources, interacting with them individually, and selecting and combining data from different sources.

The “designing data integration system” task consists of several subtasks. Ontological approach can be applied to solve two of the subtasks:

- specification of the content of heterogeneous data sources in the form of an ontology;
- answers to query that are addressed to generic system and based on source specification.

5.3.1. Specification of the content of heterogeneous data sources

The structure of data integration system allows explicitly modelling the data and information needs (i.e. to identify data that the system provides to a user) at different levels.

- The ***conceptual level*** contains a conceptual representation of data with the description of the relations between their components;
- The ***logical level*** contains a representation in terms of a logical data model of sources.

5.3.1.1. The conceptual level

The conceptual data model contains formal descriptions of concepts, relations between the data and additional requirements. These descriptions are independent from the integration system and focused on the description of the application semantics. There are three elements of the conceptual level:

- *Conceptual schema of enterprise*. General concepts related to the application are represented.
- *Conceptual schema of information source* is a conceptual data representation.
- *Domain conceptual schema* is used to describe the unification of conceptual schema of enterprise and various conceptual schemas of information sources. Relations among schemas are also included here.

Elementary concepts of relational data model such as entities, relations, and attributes, can be expressed in terms of descriptive logic.

The most interesting element at the conceptual level is the domain schema that integrates the enterprise schema and the schema of information source and links the schemas of information sources together. *Inter-scheme relations* are used.

5.3.1.2. The logical level

The logical data model provides a description of logical content of each source that is called a source schema. There are two ways to binding the source logical representation and the domain conceptual schema:

- Approach based on *global representation*. Each concept of the domain conceptual schema is associated with the basic source relations query. Thus, each concept can be understood as resource presentation;
- Approach based on local representation. The content of each basic relation within the source is expressed through concepts of the domain.

Query is used to describe the content of the source; it is a disjunction of conjunctions over the set of atoms. Each atom is a concept, a relation or an attribute.

6. Upper ontologies

Ontology is often identified with a set of classes (domain concepts) that are related to a specific set of relations. The basic types of the relations are subclass-superclass (hyponym), part-whole (meronym), instance-class, cause-effect, dependence relation, etc.

In general, ontologies are represented as consistent or shared resources: the ontology content can be simultaneously used by several people, groups, or communities. Upper ontologies include knowledge of common sense about the simulated world, forming a single system of concepts for lower-level ontologies.

6.1. OpenCyc

OpenCyc is the open source version of the Cyc technology, the world's largest and most complete general knowledge base and commonsense reasoning engine. The knowledge base OpenCyc contains information from different subject areas such as Philosophy, Mathematics, Chemistry, Biology, Psychology, Linguistics, etc. A file with OpenCyc descriptions is available for downloading at www.opencyc.com.

The key concept of OpenCyc ontology is a *collection*. Any collection can contain subcollections and instances. Thus, OpenCyc identified two taxonomic relations: subcollection - supercollection (genls) and instance – collection (isA). Any term of ontology can be an instance of the collection. An important feature of the relation isA is that it is inherited from the relation genls in hierarchy: if A is an instance of collection B, and B is a subcollection of collection C, then A is an instance of collection C. If collections A and B are connected with the relation genls (A genls B), it means all instances of collection A are instances of collection B.

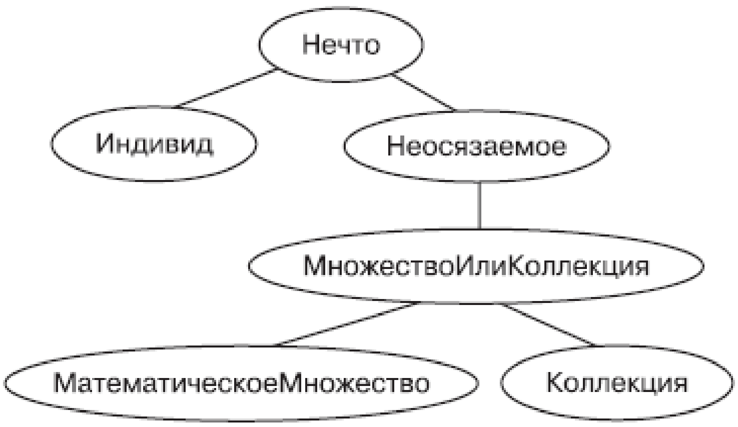


Fig. 6.1. A part of a hierarchy in OpenCyc [26]

At the top of the hierarchy is a universal collection named Thing. It contains everything that exists within the described area (so-called "universe of discourse). Any collection (Individual, MathematicalSet, or Collection) is a subcollection and instance of the collection Thing. Moreover, the collection Thing is an instance and a subcollection of itself, but not a subcollection of any other collections. At the first level of the hierarchy Thing is divided into 116 subcollections. Figure 6.1. shows a cut version of a hierarchy of upper levels.

The collection Individual contains different individuals, i.e. entities that are neither sets nor collections. Individuals can be abstract or concrete, describe physical objects, events, relations, numbers, and groups, have a complex structure, consist of parts, but no instance of this collection can have elements or subsets. Thus, an individual that has parts (part-whole relation) and a set, or a collection that contains the same parts (element – set, and instance – collection) are two different entities. For example, a firm (1), a group of all employees of the firm (2), a collection of all employees of the firm (3), and a set of all employees of the firm (4) are four different concepts. Just the first two of them are individuals.

The collection Collection contains all OpenCyc ontology collections but Thing. *Collection* is the closest concept to *class* which is often used in the developing of domain ontologies (note: it does not refer to the concept *class* of object-oriented programming!). It is because this collection describes a set of objects (instances of collection) that have some common attributes (properties). This also distinguishes Collection from MathematicalSet. A set can contain totally unrelated element, and Collection cannot. All the instances of Collection are abstract entities even though the collection contains material objects.

The knowledge base OpenCyc consists of constants (terms) and rules (formulae) that operate with these constants. The rules are divided in two types: axioms and inferable statements. Axioms in OpenCyc are statements that were clearly and manually entered in the knowledge base by experts. They did not appear there as a result of machine operations. All the statements and formulae in the knowledge base OpenCyc are recorded in the CycL language which syntax derives from first-order predicate calculus.

6.2. DOLCE

DOLCE (Descriptive Ontology for Linguistics and Cognitive Engineering) is the first module of the WonderWeb Foundational Ontologies Library. (www.loa-cnr.it/DOLCE.html)

DOLCE ontology is intended to apply in the Semantic Web for coordination among intelligent agents that use different terminology. However, DOLCE is not intended as a candidate for a “universal” standard ontology. The main purpose of the designers is to create a model that helps to compare and elucidate relations with other ontologies of the WFOL as well as to identify the hidden assumptions underlying the existing ontologies and linguistic resources such as WorldNet.

DOLCE has a clear cognitive bias, in the sense that it aims at capturing the ontological categories underlying natural language and human commonsense.

The fundamental ontological distinction between *universals* and *particulars* can be informally understood by taking the relation of *instantiation* as a primitive: particulars are entities which have no instances; universals are entities that can have instances. DOLCE is an ontology of particulars, in the sense that its domain of discourse is restricted to them. Another feature of DOLCE is a clear distinction between *enduring* and *perduring entities* (*endurants* and *perdurants*). The difference is related to their behaviour in time. Endurants are wholly present (i.e. all their proper parts are present) at any time they are present (e.g. a table, or a house during the time of their existing). Perdurants just extend in time by accumulating different temporal parts, so that, at any time they are present, they are only partially present, in the sense that some of their proper temporal parts (e.g. a hurricane or the Renaissance period) may not be present.

The definitions “object” and “process” are motivated by cognitive bias, and it does not mean that both do really exist. There are two types of relations in an ontology: *part – whole*. The first does not depend on time, while the second one has a temporal index which determines when this relation occurs. The same situation is observed in the relation quality – quality-bearer. Such basic relations of ontology as individual – process and dependence relations have a temporal index. For example, OpenCyc ontology does not have separation into endurants and perdurants. Therefore, there is no relation among a set of relations in the part “elements of objects”, which takes into account timing aspect: the possible variability of this relation.

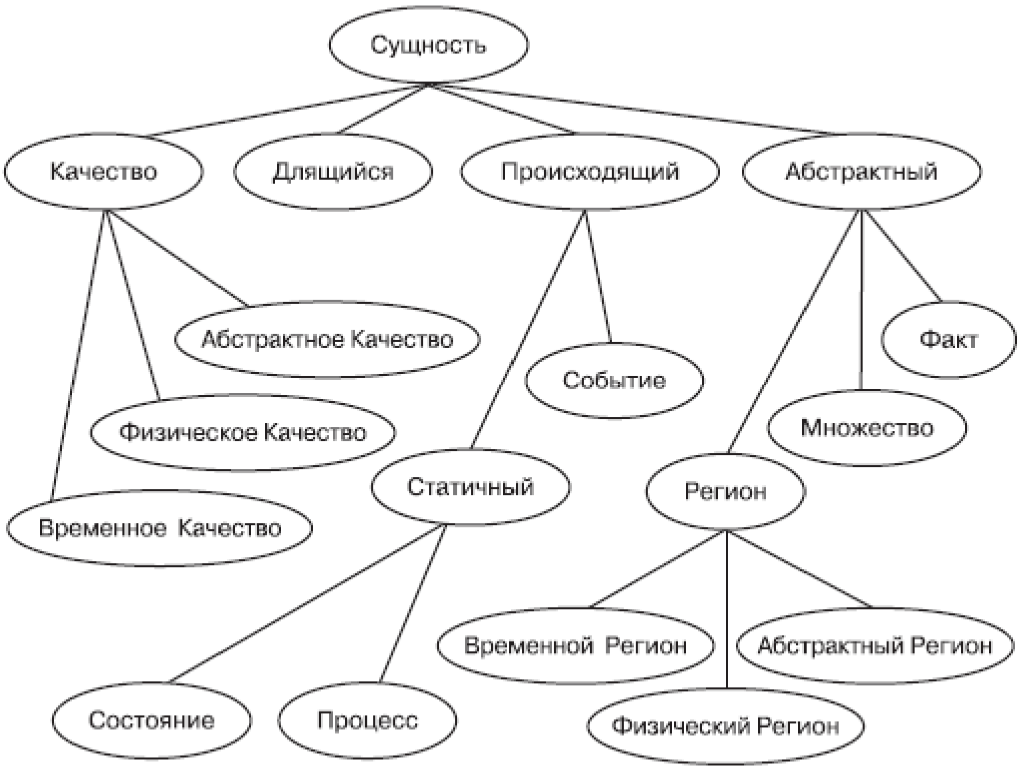


Fig. 6.2. Upper level of DOLCE hierarchy [26].

The DOLCE developers chose a more flexible approach than in the Cyc project to represent their ontology: this ontology language is based on first-order predicate logic. Then, some statements that can be presented in OWL are described. The remaining axioms are added to the OWL descriptions as comments. A disadvantage of this approach is that the applications that do nt have information about the actual structure of the OWL document cannot access the “commented” statements.

6.3. SUMO

SUMO (Suggested Upper Merged Ontology) is an upper ontology developed within the IEEE Standard Upper Ontology (IEEE SUO). (www.ontologyportal.org)

The SUMO contains general and abstract concepts, has a comprehensive hierarchy of fundamental concepts (around 1,000) and axioms (around 4,000) that define these concepts. Its purpose is to promote data interoperability, information retrieval, automated inference, and natural language processing. The ontology covers the following topics: general types of objects and processes, abstractions including set theory, attributes, and relations, numbers, and measures, temporal concepts, parts and wholes, agency and intentionality. The SUMO is an upper ontology: it contains a visible number of concepts and axioms, has an explicit hierarchy of classes, and is a result of combining various publicly available upper ontologies (including J.Sowa's ontology). The advantages of the SUMO include a possibility of translation into any basic knowledge representation language, middle-level ontology (MILO) integrated with SUMO, examples of practical use, and mapped to WordNet, the largest thesaurus so far, that contains around 150 thousand English words.

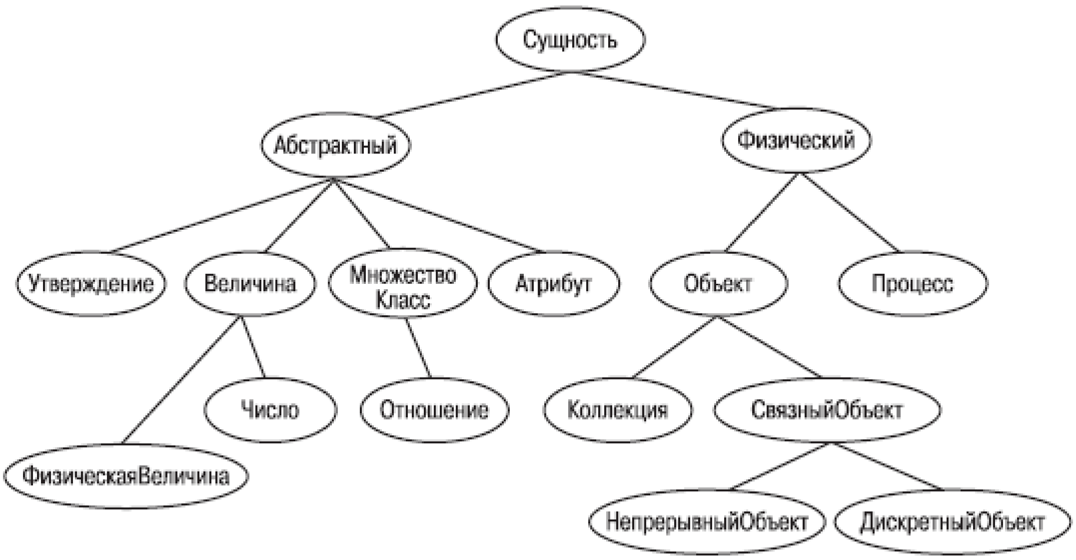


Fig. 6.3. SUMO hierarchy [26]

The SUMO hierarchy (fig. 6.3) is less complicated than in OpenCyc, and perhaps more convenient for practical use than DOLCE. The basic concept is, as in many upper ontologies, "Entity", and it subsumes "Physical" and "Abstract". The first category includes everything that has a position in space/time, and the latter one includes everything else (that exists in the mind). Under the concept of "Physical" there are disjoint concepts of "Object" and "Process" as in DOLCE.

A plane is an object, and its functioning is a process. The phrase "An aircraft lives when flying" is not just lyrics of a song but a fact of its existence, a piece of its ontology.

Immediately under the concept of "Object, there are two disjoint concepts: "SelfConnectedObject" and "Collection". A "SelfConnectedObject" is any "Object" whose parts are all mediately or immediately connected with one another.

The concept of “SelfConnectedObject” is portioned into two concepts: “ContinuousObject” and “CorpuscularObject”. A “ContinuousObject” is an “Object” in which every part is similar to every other in every relevant respect. More precisely, something is a “ContinuousObject” when all of its parts (down to an unspecified level of granularity) have the properties of the whole. Thus, such substances as water and clay would be subslasses of the concept of “ContinuousObject” as would topographic locations like surface and geographical areas.

“Collections” are disjoint from “SelfConnectedObjects”. “Collections” consist of disconnected parts, and the relation between these parts and their corresponding “Collection” is known as “member” in the SUMO. As in OpenCyc, the concepts of “Collection”, “Class” and “Set” are separated. A “member” predicate is different from the “instance” and “element” predicates, which relate things to the “Classes” or “Sets” to which they belong. Unlike “Classes” and “Sets”, “Collections” have a position in space-time, and “members” can be added and subtracted without thereby changing the identity of the “Collection”. Some examples of “Collections” are toolkits, design firms, and an aircraft park.

The class “Abstract” subsumes four disjoint concepts: “Set”, “Proposition”, “Quantity”, and “Attribute”.

- “Set” is the ordinary set-theoretic notion, and it subsumes “Class”, which, in turn, subsumes “Relation”. A “Class” is understood as a “Set” with a property or conjunction of properties that constitute the conditions for membership in the “Class”, and a “Relation” is a “Class” of ordered tuples. “Relation” is immediately subsumed by “Class”, rather than “Set”, because “Relation” is restricted by those ordered tuples that express its content.
- The concept of “Proposition” corresponds to the notion of semantic or informational content. The Sumo places no size restrictions on this content. This is a broader notion than it is used in many ontologies. Some examples of “Propositions” would be a short description of a project, a performance review.
- The class of “Attributes” includes all qualities, properties, etc. that are not reified as “Objects”. For example, rather than dividing the class of “Animals” under “Objects” into “FemaleAnimales” and “MaleAnimales”, “Female” and “Male” instances of “BiologicalAttribute”, a subclass of “Attribute”, are made. The same is about civil and military airplanes.
- “Quantity” is divided into “Number” and “PhysicalQuantity”. The first is understood as a count independent of an implied or explicit measurement system, and the latter is taken to be a complex consisting of a “Number” and a particular unit of measure.

The axioms limit the interpretation of the concepts and provide a foundation for a computer-aided reasoning, which can process relevant to the SUMO structure knowledge bases. An example of an axiom: “If C is an instance of *combustion*, then there are such *heat emission* H and *light emission* L as they both (H and L) are subprocesses of C”. Thus, the processes of *heat emission* and *light emission* accompany each *combustion process*.

6.4. J.Sowa's ontology

J.Sowa's ontology, described in the book "Knowledge Representation: Logical, Philosophical, and Computational Foundations", defines the basic categories derived from a variety of sources in logic, linguistics, philosophy, and artificial intelligence. (www.jfsowa.com/ontology)

To keep the system open-ended, the ontology, according to Sowa, is not based on a fixed hierarchy, but on a framework of distinctions, from which the hierarchy is generated automatically. For any particular application, the categories are not defined by drawing lines on a chart, but by selecting an appropriate set of distinctions.

Table 6.1. Ontological categories of the upper level by J.Sowa

	Physical		Abstract	
	Continuant	Occurrent	Continuant	Occurrent
Independent	Object	Process	Schema	Script
Relative	Juncture	Participation	Description	History
Mediating	Structure	Situation	Reason	Purpose

Table 6.1. shows the ontological categories. The important notions are: ***entity***, that is the default type for anything of any category, it does not define any features and differences; and the ***absurd type***, which inherits all differentiae. It is not possible for any existing entity to be an instance of the absurd type. The distinction between abstract and physical is made in the ontology (in this form it is taken by developers of the SUMO). There are categories: independent, relative, and mediating. An "independent" entity does not need any relationship to anything else. Any "relative" entity must have some relationship to something else, and a "mediating" entity creates a relationship between two other entities. John Sowa's ontology describes roles and relations, agents, processes, etc.

7. Domain ontology

7.1. CIDOC CRM ontology

CIDOC CRM ("Committee on Documentation Conceptual Reference Model") is a formal ontology intended to facilitate the integration, mediation and interchange of heterogeneous cultural heritage information. More specifically, it defines and is restricted to the underlying semantics of database schemata and document structures used in cultural heritage and museum documentation in terms of a formal ontology. It does not define any of the terminology appearing typically as data in the respective data structures; however, it forces the characteristic relationships for its use.

It is intended to be a common language for domain experts and is implemented to formulate requirements for information systems and to serve as a guide for good practice of conceptual modelling. It can provide the "semantic glue" needed to mediate between different sources of cultural heritage information, such as that published by museums, libraries and archives.

CIDOC CRM consists of a hierarchy of classes and a wide range of properties. A class may be the domain or the range of properties, which are binary relations among classes. All concepts (classes and properties) of the model can be divided into three groups. The first group includes the classes and relations that cover the most general concepts of the world: both permanent and temporary entities, relations, matching in time. The second one contains the concepts that partially support managerial functions: accounting and purchasing items, the transfer of ownership on cultural objects. The third group includes classes and properties used for the internal organisation of the ontology itself: the resources to connect external sources of terms, such as thesauri by cultural fields.

The CIDOC CRM model is divided into *permanent* entities and *temporal* entities. The others are subsidiary classes.

On figure 7.1. the different parts of CIDOC CRM ontology (the screen shots are made using the Protégé editor) are presented [26].

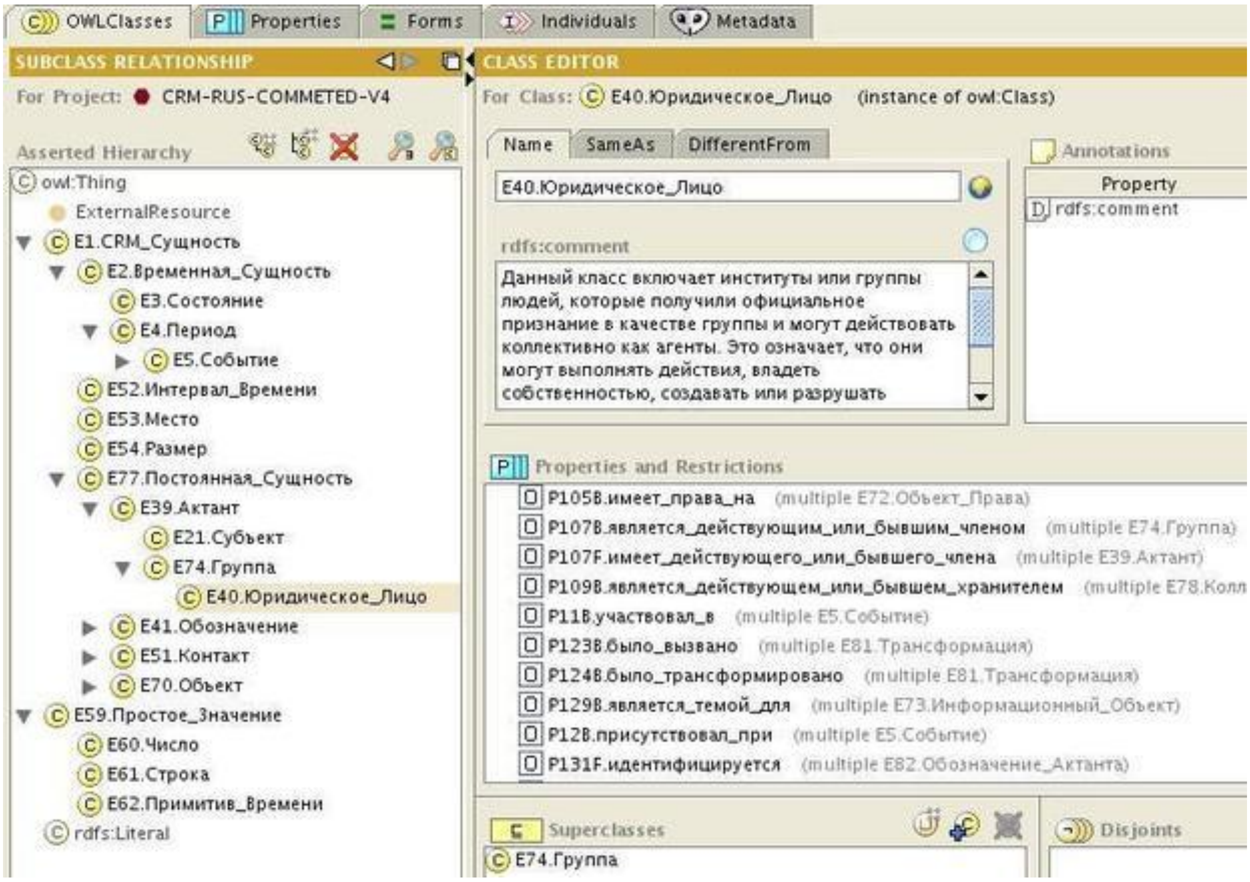


Fig. 7.1. The hierarchy of classes of the CIDOC CRM ontology. Text description and properties of the class "Юридическое_Лицо" (Legal Entity).

7.2. Products and Services Ontology

One of the important application areas of ontology is products and services. The tasks that ensure the products and services ontology include:

- Gather information about the products;
- Presentation of information about the product;
- Products classification: separation by purpose;
- Search by products;
- Banners display;
- Text information about products (reviews, news, forums).

Rubricators are generally used to classify products. Rubricators are considered as a kind of ontological knowledge organisation.

7.2.1. Rubricators as an ontology

Rubricator is a classification table of hierarchical classification that contains the full list of classes of the system and is intended to organise information collections, arrays, and publications, and to search among them (GOST 7.74-96).

The main difference between the terms of a thesaurus and rubrics of a rubricator is that the terms are fundamentally linguistic notions, while the rubrics correspond to the conceptual categories. The purpose of developing an information retrieval thesaurus is to find good, short words and phrases to describe the main topics of the documents bringing synonyms and quasi synonyms to thesaurus' descriptors. [26]

The purpose of the rubricator is to develop separate disjoint conceptual categories. Ideally, there should be no intersection between the rubrics, and no gaps, that is all subdomains should be within the rubrics of the rubricator. To achieve these goals, the rubricator is structured. There are two ways to do this: hierarchical organisation and faceted organisation. To specify the rubrics and avoid intersections of values, it is often required to give long names to the rubrics. It is done to determine a separate conceptual category and avoid finding this name in a thesaurus. Since working with these rubrics is difficult enough, a code classification system is usually assigned to them.

7.2.2. OntoSeek

OntoSeek is a system designed for content-based retrieval in such publications as “Yellow Pages” and product catalogues.

The features of the system are:

- The option to use arbitrary natural-language terms for goods and services description;
- Complete terminological flexibility for the queries, due to a process of ontology-driven semantic matching between queries and resource descriptions;
- Interactive assistance on query formulation, generalisation, or specialisation.

To represent queries and resource description conceptual graphs were selected. Compared with simple attribute-value lists, they are much more flexible and significantly more expressive. With conceptual graphs, the problem of content marching reduces to ontology-driven graph matching.

The WordNet ontology is taken; based on which descriptions synonymy and genus-species relationship can be identified.

The construction of requests is based on graphs that contain variables. For example, if a user is looking for cars with radio, then the request will look like this:

[<X> car]-> (part) -> [Radio].

If a user is looking for a radio for a car, then the request will be presented as:

[car]-> (part) -> [<X> Radio].

There is a problem in such ontologies as WordNet related to a lack of ontological information about mutual disjointness between concepts. To solve this problem it was proposed to distinguish concepts-types and concepts-roles and to introduce the following assumptions:

- Types of another hierarchy rather than the genus-species one are mutually exclusive;
- Roles are always subject to types;
- Roles that are subject to the same type are considered as not mutually exclusive unless specifically indicated, for example, via the antonym link.

Examples of types are person and plant, while student and child are examples of roles. Types and roles differ basically in that types are intrinsically essential properties – their instances necessarily belong to them. This is not the case for roles because a student can cease to be a student and still remain the same individual.

8. Ontology description languages

8.1. Metadata architecture for the World Wide Web

8.1.1. Resources

The thing which you get when you follow a URL link can be called a resource. Sometimes it is referred to as a document or an object, when the object is something which is more machine readable in nature or has a hidden state. Within this chapter the words ‘document’, ‘resource’, and ‘object’ are synonyms.

One of the characteristics of the World Wide Web is that resources, when they are retrieved, do not stand simply for themselves without explanation, but there is information about the resource. Information about information is generally known as *Metadata* that is machine understandable information about web resources or other things. The phrase “machine understandable” means information which software agents can use in order to do the tasks that a user requested. Metadata has well defined *semantics* and *structure*. Therefore, the Web has data, which is resources, and metadata, information about resources.

8.1.2. The form of Metadata

Metadata consists of assertions about data, and such assertion typically, when represented in computer system, take the form of a name or type of assertion and a set of parameters.

The most common form of expression is the model Resource – attribute – value. *Resource* is an object about which an expression is set; *attribute* is a property or parameter of object; *value* is a value of attribute.

8.1.3. Links

Links are the basis of navigation in the World Wide Web. They can be used for building structure within the Web and also for creating a Semantic Web which can express knowledge about the world itself. That is to say, links may be used both for structure of data, in which case they are metadata, but also they may be used as a form of data.

One of the main tasks of the designing of metadata architecture for the Web is to develop self-describing information. An advantage of self-describing information is that it allows development of new applications and new functionality independently by many groups across the web. Without self-describing information, development must wait for large companies or standards committees to meet and agree on the commonly agreed semantics.

8.1.4. RDF

The Resource Description Framework (RDF) is a language for representing information in the World Wide Web. It is particularly intended for representing metadata about Web resources, such as the title, author, etc. RDF can be used to represent information about anything that can be identified on the Web (via URI).

8.1.5. RDF graph

The RDF model is based on sets of statements or triples, each of which consists of subject, predicate, and object (S, P, O). This set of triples is called RDF graph (fig. 8.1). The source node of an edge of the directed graph is the subject of the statement and the target is the object. The edge is labelled with the predicate.

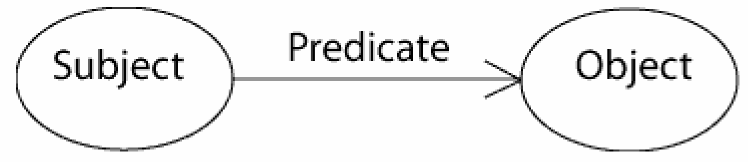


Fig. 8.1. An RDF triple

Each triple represents a statement of a relationship between the things denoted by the nodes that it links (S, P, and O). Subject and predicate are URI references. Object is a URI reference, or a RDF-literal.

8.1.6. RDF-literals

Literals may be plain or typed. A literal in a RDF graph contains one or two named components. All literals have a lexical form being a Unicode string. Simple (plain) literals have a lexical form and optionally a language tag (ru, en). Typed literals have a lexical form and a datatype URI being an RDF URI reference.

8.1.7. Literal Equality

Two literal are equal if and only if all of the following hold:

- The strings of the two lexical forms compare equal, character by character.
- Either both or neither have language tags.
- The language tags, if any, compare equal.
- Either both or neither have datatype URIs.
- The two datatype URIs, if any, compare equal, character by character.

8.2. Ontology languages

When developing an ontology it is important to choose an appropriate ontology specification language. The purposes of such languages are to specify the additional machine-interpretable semantics of resources, to make the machine data interpretation more like in the real world, to greatly increase the possibilities of conceptual modelling of weakly-structured Web-data.

Distribution of the ontological approach to knowledge representation has assisted in the development of ontology languages and the tools for their editing and analysis. There are ontology specification languages: Ontololingua, CycL, description logic languages (such as LOOM), frame-based languages (OKBC, OCML, F-Logic). The newer languages are based on Web-standards (XOL, SHOE, UPML). Languages RDF, RDFS, DAML+OIL, OWL were developed in order to exchange ontologies in the Web.

The languages that describe metadata and ontology in the Web are well-developed: query languages and rule languages are brought to the level of technological standards in the domain. However, the weakest link is still the mechanism of interaction between agents based on ontology.

8.2.1. RDFS

RDF does not provide any mechanism for describing the attributes of resources and defining relationships between them. That is the role of the RDF vocabulary description language, RDF Schema. RDFS defines and properties that may be used to describe classes, properties and other resources.

RDF Schema is a semantic extension of RDF. It provides mechanism for describing groups of related resources and the relationships between these resources. RDFS vocabulary descriptions are written in RDF using the terms described in this document. These resources are used to determine characteristics of other resources, such as the domains and ranges of properties.

The RDF vocabulary description language class and property system is similar to the type systems of object-oriented programming languages such as Java. RDF differs from many such systems in that instead of defining a class in terms of the properties its instances may have, the RDF vocabulary description language describes properties in terms of the classes of resource to which they apply. This is the role of the domain and range mechanisms described in this specification. Using the RDF approach, it is easy for others to subsequently define additional properties. This can be done without the need to re-define the original description of these classes. One benefit of the RDF property-centric approach is that it allows anyone to extend the description of existing resources.

Resources may be divided into groups called classes. The members of a class are known as *instances* of the class. Classes are themselves resources. They are often identified by RDF URI References and may be described using RDF properties. The rdf:type property may be used to state that a resource is an instance of a class.

RDF distinguishes between a class and the set of its instances. Associated with each class is a set, called the class extension of the class, which is the set of the instances of the class. Two classes may have the same set of instances but be different classes.

In the case when it is necessary to make a statement on the approval of RDF, reification, or the materialisation of the allegations, takes place. In this case, the statement itself (or statement) serves as an object.

The openness and extensibility of RDF is to ensure that "anyone (i.e. any RDF user) can say anything (i.e., fix an arbitrary statement) about anything (i.e. about any resource)," using RDF. RDF does not prohibit making meaningless statements or statements that are not consistent with the others. Therefore, there is no guarantee of integrity and consistency of RDF-descriptions. The responsibility for verifying lies on the recipient of metadata, i.e. on application developers that process RDF-data.

8.2.2. OWL

The OWL (Web Ontology Language, an acronym, the letters are swapped intentionally to get the word “owl”) is an ontology representation language. In fact, it is a dictionary that extends the set of terms defined by RDFS. An OWL ontology may include descriptions of classes, properties and their instances. The language

DAML + OIL, bringing together two initiatives: the DAML (DARPA Agent Markup Language) project and the OIL (Ontology Inference Layer) project, was the predecessor of OWL. The SHOE (Simple HTML Ontology Extensions) is the earliest Web ontology representation project. The development of Web ontology description languages is shown in Fig. 8.2. The upper level representatives, OIL, DAML + OIL and OWL, are still evolving, but OWL is the most popular.

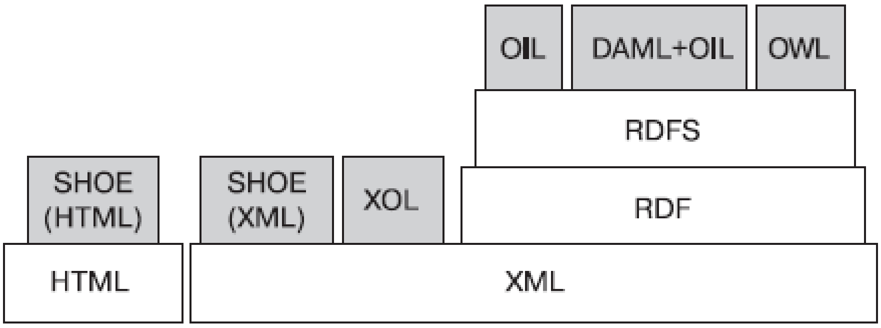


Fig. 8.2. Web ontology description languages [26]

The OWL language has three sublanguages:

- OWL Lite has the least expressive power but for simple tasks it might be enough. OWL has an important property – the solvability, i.e. the problem of drawing the conclusion from statements formulated in this language is computable;
- OWL DL has more expressive power than OWL Lite. It also has the property of solvability; however, the computational complexity is higher than in OWL Lite. This property can be achieved by putting restrictions on the syntax of the language. In OWL DL a class cannot be an instance.
- OWL Full is the most expressive language. It is equal to RDF. When using OWL Full there are no guarantees on the computability conclusions.

Each of these languages (except OWL Lite) is an extension of the previous one. Therefore, every OWL Lite ontology is an OWL DL ontology, and every OWL DL ontology is an OWL Full ontology.

The structure of OWL ontology has a header and a body. The header contains information about the ontology (version, notes) and imported ontologies. The header is followed by the body of ontology that contains descriptions of classes, properties and instances.

A new term, class (owl:Class) is introduced in OWL. It is because not all classes of the sublanguages OWL DL and OWL Lite are classes of RDFS (in this case owl:Class is a subclass of rdfs:Class). The sublanguage OWL Full does not have such restrictions, and owl:Class is in fact a synonym for rdfs:Class.

The property rdfs:subClassOf is used to organise classes into hierarchy.

Owl:Thing and owl:Nothing are two complementary classes. The first one is a superclass of any class of OWL, and the second class is a subclass of any class of OWL. An instance of any class of OWL is included into an extension of the class owl:Thing. An extension of the class owl:Nothing is an empty set.

OWL distinguishes six types of class description:

1. a class identifier (a URI reference)

2. an exhaustive enumeration of individuals that together form the instances of a class
3. a property restriction
4. the intersection of two or more class descriptions
5. the union of two or more class descriptions
6. the complement of a class description.

The first type is special in the sense that it describes a class through a class name (syntactically represented as a URI reference). The other five types of class descriptions describe an anonymous class by *placing constraints on the class extension*.

Class descriptions of type 2-6 describe, respectively, a class that contains exactly the enumerated individuals (2nd type), a class of all individuals which satisfy a particular property restriction (3rd type), or a class that satisfies Boolean combinations of class descriptions (4th, 5th and 6th type). Intersection, union and complement can be respectively seen as the logical AND, OR and NOT operators.

OWL classes are described through “class description”, which can be combined into “class axioms”.

Class descriptions form the building blocks for defining classes through class axioms. The simplest form of a class axiom is a class description of type 1.

<owl:Class rdf:ID="Самолет"/>

It just states the existence of the class “Самолет”.

OWL contains three language constructs for combining class descriptions into class axioms:

- rdfs:subClassOf allows one to say that the class extension of a class description is a subset of the class extension of another class description.
- owl:equivalentClass allows one to say that a class description has exactly the same class extension as another class description.
- owl:disjointWith allows one to say that the class extension of a class description has no members in common with the class extension of another class description.

OWL distinguishes between two main categories of properties: object properties that link individuals to individuals, and datatype properties, that link individuals to data values.

An object property is defined as an instance of the built-in OWL class owl:ObjectProperty. A datatype property is defined as an instance of the built-in OWL class owl:DatatypeProperty. Both are subclasses of the RDF class rdf:Property.

A property axiom defines characteristics of a property. In its simplest form, a property axiom just defines the existence of a property. For example

<owl:ObjectProperty rdf:ID="hasParent"/>

This defines a property with the restriction that its values should be individuals.

Often, property axioms define additional characteristics of properties. OWL supports the following constructs for property axioms:

- RDF Schema constructs: rdfs:subPropertyOf, rdfs:domain and rdfs:range;

- relations to other properties: owl:equivalentProperty and owl:inverseOf;
- global cardinality constraints: owl:FunctionalProperty and owl:InverseFunctionalProperty;
- logical property characteristics: owl:SymmetricProperty and owl:TransitiveProperty.

Individuals are defined with individual axioms (also called "facts"). Consider two types of facts:

1. Facts about class membership and property values of individuals
2. Facts about individual identity

An example of the first type individuals:

<Самолет rdf:ID="Ту-154">
 <имеетКонструктора rdf:resource="#Туполев"/>
</Самолет>

This axiom includes two facts: 1. The individual Ту-154 (Tu-154) is an instance of the class Самолет (Aircraft), 2. This individual is linked with individual Туполев (Tupolev) (which is defined somewhere else) by the property имеетКонструктора (hasConstructor). The first fact states a “membership” in the class, and the second one talks about the properties of the individual.

Many languages have a so-called "unique names" assumption: different names refer to different things in the world. On the web, such an assumption is not possible. For example, the same person could be referred to in many different ways (i.e. with different URI references). For this reason OWL does not make this assumption. Unless an explicit statement is being made that two URI references refer to the same or to different individuals, OWL tools should in principle assume either situation is possible.

OWL provides three constructs for stating facts about the identity of individuals:

- owl:sameAs is used to state that two URI references refer to the same individual.
- owl:differentFrom is used to state that two URI references refer to different individuals
- owl:AllDifferent provides an idiom for stating that a list of individuals are all different.

9. Ontology building tools

Applications specially created for designing, editing and analysis of ontologies are called *ontology editors*. The main function of any ontology editor is to maintain the process of knowledge formalisation and to represent an ontology as a specification.

The number of publicly available editors is more than 100. However, the quantity has not become quality yet. The table 7.1. shows the main characteristics of the most popular ontology editors.

Table 7.1. Comparison of ontology editors

Name	**Descriptions**	**Formalisms, languages, formats**	**URL**
Ontolingua	Collaborative development of ontologies	OKBC, KIF	www.ksl.stanford.edu/software/ontolingua/
Protégé	Creation, visualisation and manipulation of ontologies	JDBC, UML, XML, XOL, SHOE, RDF/RDFS, DAML+OIL, OWL	protege.stanford.edu
Magenta	Ontology constructor		http://www.magenta-technology.com/
OntoSaurus	A Web-browser for Loom knowledge bases	LOOM	www.isi.edu/isd/ontosaurus.html
OntoEdit	Ontology development and maintenance	F-Logic, RDFS, OIL, OXML	www.ontoknowledge.org/tools/ontoedit.shtml
OilEd	Ontology constructor	DAML+OIL	oiled.man.ac.uk
WebOnto	Ontology development	OCML	kmi.open.ac.uk/projects/webonto/
WebODE	Ontology development with Methontology	F-Logic, LOOM, Ontolingua	webode.dia.fi.upm.es/WebODEWeb/index.html

9.1. Formalisms and representation formats

Formalism is a theoretical basis underlying ontology knowledge representation. Examples of formalisms are First-Order Logic FOL, descriptive logic DL, Frames, conceptual graphs, etc. The formalism used by an editor can significantly affect the internal data structures and define the representation format, or a user’s interface.

The format of ontology representation specifies the type of storage and transferring of ontological descriptions. The format stands for ontology representation languages: RDF, OWL, etc.

Ontology editors usually support several formalisms and representation formats. However, in most of the cases just one formalism is native for a particular editor.

9.2. Functionality of ontology editor

The functionality of editor is considered as a set of scripts that describe how to use it. The basic set of functions ensures:

- Work with one or more projects:
- saving a project in the right formalism and format (export);
- opening a project;
- import from an external format;
- editing the project metadata (from the settings of editing form and data representation to the support of project version).
- Ontology editing (creating, editing, deleting of concepts, relations, axioms and other structural elements of ontology, editing of taxonomy).

Additional features of editors include query language support (for non-trivial assertion retrieval), integrity analysis, the use of inference method, multi-user support, service for remote Internet access.

The required tools for entering and editing ontological information:

- ***Alignment (matching)***: the process of determining correspondences between concepts of ontologies;
- ***Mapping***: the process of finding semantic relationships between entities of two ontologies;
- ***Merging***: the act of bringing together two conceptually divergent ontologies or the instance data associated to two ontologies.

9.3. Main ontology editors

Ontolingua

In addition to the ontology editor, the Ontolingua Server includes a Webster (to obtain terms definitions), an OKBC (Open Knowledge Base Connectivity) server (to access Ontolingua ontologies through the OKBC protocol), and Chimaera (to analyse, merge and integrate ontologies).

DOE (Differential Ontology Editor)

The specification process is divided in three steps:

1. The user is invited to build taxonomies of concepts and relations, explicitly justifying the position of each item (notion) in the hierarchy. Then, the user has to explicit why a notion is similar but more specific than its parent. The user can also add synonyms and encyclopaedic definition in a few languages for all notions.
2. The two taxonomies are considered from an extensional semantics point of view. The user can augment them with new entities (defined) or add constraints onto the domains of the relations.
3. The ontology can be translated into a knowledge representation language.

OntoEdit

It is a tool that provides visualisation, check and modification of ontology. It supports the representation languages OIL and RDFS as well as internal knowledge representation language OXML, based on XML. It is an autonomous Java application but its codes are closed. OntoEfit Free is limited by 50 concepts, 50 relations, and 50 instances.

OilEd

It is an autonomous graphics ontology editor developed within the On-To-Knowledge project. It is available as an open source project under the GPL license. The tool uses OIL language for ontology representation. OilEd does not have support of instances of classes.

WebOnto

WebOnto is a Java applet. It was designed for visualisation, browsing and editing support for developing and maintaining ontologies and knowledge models specified in OCML (Operational Conceptual Modelling Language). A user can create different structures including classes with multiple inheritance.

ODE, WebODE

ODE (Ontological Design Environment) interacts with users on a conceptual level. It provides a number of tables (to fulfil with concepts, attributes, relations) and automatically generates codes in LOOM, Ontolingua and F-Logic languages. This tool was developed within the ontology editor WebODE which integrates the ODE services into one architecture keeping its ontology in a relational database.

Magenta

It is an ontology constructor and an executive system of MAGENTA CORPORATION LIMITED. It is presented in details in the guidelines [6].

Protégé

This is a free, open source Java programme designed to build (create, edit and view) domain ontologies. It includes an ontology editor that helps to make ontologies about a domain by just expanding a hierarchical structure, and including abstract or concrete classes and slots. Based on the ontology built, Protégé is able to generate a knowledge acquisition tool for entering the instances of the ontology.

Initially Protégé was used by experts for conceptual modelling in medicine. Lately, it is applied in other domains, in particular for building ontologies in Semantic Web.

In the beginning Protégé was a frame based model. This formalism is considered as its own for the editor but not the only one.

Protégé has an open and easily expandable architecture. In addition to frames it supports the most popular knowledge representation languages (SHOE, XOL, DAML+OIL, RDF/RDFS, OWL) as well as plug-ins. Protégé is based on the OKBC knowledge representation. The main elements are classes, instances, slots (which represent class and instance properties), and facets (set additional information on slots).

The user's interface consists of the main menu and several tabs for editing different parts of knowledge base and its structure. The tabs depend on the project type (representation language) and can be manually adjusted. Usually there are the tabs as follows: classes, slots (or OWL properties), instances, metadata. The function of the main tabs is to provide a set of forms for filling in the knowledge base.

The functions of the tab "***Classes***": creating classes, slots for this class, class hierarchy visualisation, adding next notes to the classes, class retrieval by sample.

The functions for the tab “***Slots***”: creating slots, domain defining, hierarchy and slot properties visualisation, adding slots’ text description, slot retrieval by sample, setting the slot value limits.

The functions of the tab “***Instances***”: creating instances of the class, instance properties visualisation and editing, class hierarchy visualisation, adding text description of instances, slot retrieval by sample, setting the slot value limits.

9.4. Methodology for building ontologies

In practical terms, developing an ontology includes:

- Defining classes in the ontology;
- Arranging the classes in a taxonomic (subclass-superclass) hierarchy;
- Defining slots and describing allowed values for these slots;
- Filling in the values for slots for instances.

After that, a knowledge base can be created by defining individual instances of these classes filling in specific slot value information and additional slot restrictions.

Fundamental rules in ontology design:

1. There is no one correct way to model a domain – there are always viable alternatives. The best solution almost always depends on the application that one have in mind and the anticipated extensions;
2. Ontology development is necessarily an iterative process;
3. Concepts in the ontology should be close to objects (physical or logical) and relationships in the domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe the domain.

That is, deciding what the ontology is going to be used for, and how detailed or general the ontology is going to be will guide many of the modelling decisions down the road. Among several viable alternatives, it will be needed to determine which one would work better for the projected task, be more intuitive, more extensible, and more maintainable. It is important to keep in mind that an ontology is a model of reality of the world and the concepts in the ontology must reflect this reality.

After an initial version of the ontology is defined, it can be evaluated and debugged by using it in applications or problem-solving methods or by discussing it with experts in the field, or both. As a result, the initial ontology almost certainly will be required to revise. This process of iterative design will likely continue through the entire lifecycle of the ontology.

Reusing existing ontologies may be necessary if the system needs to interact with other applications that are already included in the individual ontologies or controlled vocabularies. Many useful ontologies are already available in electronic form and can be imported. An example from the aviation is the aero.owl ontology, available for its development in the editor Protégé.

References

1. Боргест Н.М. Автоматизация предварительного проектирования самолета. Учеб. пособие. Самара: САУ, 1992 – 92 с.
2. Боргест Н.М. Антология онтологии/подборка научных статей. – Самара: СГАУ, 2010 – 88 с.
3. Боргест Н.М. Истоки и источники онтологии проектирования/Список рекомендуемой литературы и интернет-ресурсов. Самара: СГАУ, 2010 – 16 с.
4. Боргест Н.М., Комаров В.А., Данилин А.И. Краткий словарь авиационных терминов/ Под ред. Комарова В.А. – М.: Изд-во МАИ, 1992. – 224 с.
5. Боргест Н.М., Симонова Е.В. Основы построения мультиагентных систем, использующих онтологию. Учеб. пособие. Самара: СГАУ, 2009 – 80 с.
6. Боргест Н.М., Симонова Е.В., Шустова Д.В. Решение проектных задач с помощью онтологических систем/ Метод. указания к лаб. раб. – Самара: СГАУ, 2010 – 128 с.
7. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.
8. Дембицкий Н.Л., Назаров А.В. Применение методов искусственного интеллекта в проектировании и производстве радиотехнических устройств: Монография. – М.: Изд-во МАИ-ПРИНТ, 2009. – 228 с.
9. Евгенев Г.Б. Интеллектуальные системы проектирования: учебное пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2009. – 334 с.
10. Ездаков А.Л. Экспертные системы САПР: учебное пособие. – М.: ИД «ФОРУМ», 2009. – 160 с.
11. Колчин А.Ф., Овсянников М.В., Стрекалов А.Ф., Сумароков С.В. Управление жизненным циклом продукции. – М.: Анахарсис, 2002. – 304 с.
12. Комаров В.А., Боргест Н.М. и др. Концептуальное проектирование самолета. Учеб. пособие. Самара: СГАУ, 2007 – 92 с.
13. Муромцев Д.И. Онтологический инжиниринг знаний в системе Protégé. – СПб: СПб ГУ ИТМО, 2007. – 62 с.
14. Мышкин Л.В. Прогнозирование развития авиационной техники: теория и практика. – М.: ФИЗМАТЛИТ, 2008. – 328 с.
15. Проектирование самолетов: учебник для вузов/С.М. Егер, В.Ф. Мишин, Н.К. Лисейцев и др. Под ред. С.М. Егера. Науч. предисловие А.М. Матвеенко, М.А. Погосяна, Ю.М. Шустрова. – 4-е изд. Репр. воспр. текста изд. 1983 г. - М.: Логос, 2005. - 648 с.
16. Пупков К.А., Коньков В.Г. Интеллектуальные системы. – М.: Изд-во МГТУ им. Н.Э.Баумана, 2003. - 348 с.
17. Рапопорт Г.Н., Герц А.Г. Искусственный и биологический интеллекты: Общность структуры, эволюция и процессы познания. М.: КомКнига, 2010. – 312 с.

18. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. – М.: Вильямс, 2007. - 1408 с.
19. Редько В.Г. Эволюционная кибернетика. – М.: Наука, 2003. – 156 с.
20. Тейлор Д., Рэйден Н. Почти интеллектуальные системы. Как получить конкурентные преимущества путем автоматизации принятия скрытых решений. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 448 с.
21. Уитби Б. Искусственный интеллект: реальна ли Матрица. – М.: ФАИР-ПРЕСС, 2004. – 224 с.
22. Шапиро Д.И. Виртуальная реальность и проблемы нейрокомпьютинга.- М.: РФК-Имидж Лаб, 2008. - 454 с.
23. Шведин Б.Р. Онтология предприятия: экспириентологический пожход. Технология построения онтологической модели предприятия. М.: ЛЕНАНД, 2010. -240 с.
24. Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка эксперных систем. Среда CLIPS. – СПб.: БХВ_Петрбург, 2003. – 608 с.
25. A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.2, Nick Drummond, Simon Jupp, Georgina Moulton, Robert Stevens. - The University Of Manchester: March 13, 2009
26. http://www.intuit.ru/department/expert/ontoth/ Онтологии и тезаурусы: модели, инструменты, приложения: курс лекций. - Б.В. Добров, В.В. Иванов, Н.В. Лукашевич, В.Д. Соловьев
27. http://ru.wikipedia.org/wiki/Онтология_(информатика)
28. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html
29. http://www.masters.donntu.edu.ua/2010/fknt/bolotova/library/tez1.htm Болотова В.А. Григорьев А.В., Инструментальные средства создания баз знаний на основе системы онтологий.
30. http://www.masters.donntu.edu.ua/2010/fknt/bolotova/library/tez3.htm Григорьев А.В. Формирование прикладных онтологий.
31. http://www.aviaslovar.ru Краткий словарь авиационных терминов.
32. http://protege.stanford.edu/ Сайт редактора онтологий Стэндфордского университета
33. http://www.magenta-technology.com/ Сайт компании Magenta Corporation Limited.
34. http://www.sociograd.ru/index.php/students/17-news/137-savchenkoiv Леньков Р.В. Социальное предвидение в управлении: феномены, формы, проблемы
35. http://www.shkp.ru/lib/archive/second/2001-1/1 Генисаретский О.И.Лекция о месте проектирования в системе стратегической работы
36. http://www.philosophy.ru/iphras/library/filtech.html Философия техники. История и современность.