# What are to be recognized on the screen? Usability Reference Model

Gábor Vitályos

Vitályos Consuting, Budapest, Hungary
consulting@vitalyos.hu

[…] there aren't yet correct references in this version of the paper.

***Abstract*- The lack of the strict conceptual basis, the terminological disorders and misunderstanding, the difficulties of the documentation works, these all obstruct more and more the portal building work in all of its phases: in the design, the implementation, the use, the support (e.g. to understand the user reflections), the audit, and in the training.**

**In this paper we continue the research on the requirement, on the usability of the professional e-services. It is an important chapter of the Cognitive Info-Communication sciences, because the client deeds to recognize without effort and waste of time what he/she sees on the screen: the structure of the portal, the content table, the data panels, the forms, the dialogue states.**

**There are many experimental researches on the behavior of the users: the eye tracking, browse tracking, habits of searching, etc. In this paper we address theoretical investigation of behavior of the portals, from the point of view of the HCI.**

**Although we are over-packed with CMS software technologies, none of them can be thought of as a technically thought-out, theoretically established, exact technology.**

**Our analysis is mainly conceptual or semantic and led to a Usability Reference Model, which gives us a map of notions. Based on it, we can build a strict conceptual basis, a nomenclature for the usability discipline, avoid the subjectivity, and we formulate the requirements of the usable portal.**

**The paper mainly addresses the conventional ICT technology, namely desktop or mobile PC-s, smart phones, for this is the area of the conventional e-services. However, we hope that our results can be valid even in the virtual and extended reality also which nowadays is a mainstream area of Cognitive Info-Communication.**

## I. Introduction

*Keywords*: e-government, HCI (Human Computer Interaction), usability, IA (Information Architecture), software technology, Internet, ontology, OO (Object Oriented) technology, DFU (Design for Usability), CMS.

***Terminology***

*Portal*: 'electronic content', 'e-content', 'electronic service', 'e-service', 'Internet service' are used here as equivalent terms. We use the term *portal* for all of them.

*URM* (Usability Reference Model): model, elaborated and published in this paper. Its more correct name is *IARM* (Information Architecture Reference Model), and for the simplicity we use the URM.

*URM philosophy:* we often refer that '*URM philosophy* requires'. The consistent summarization of this philosophy and its 'requirements' will be in a subsequent paper.

*IConS* (Interactive Contents & Services): the term for work of the planned, hypothetic new SMC technology complying with the URM our requirements.

*Virtual space* and *object*: the structure of a *portal*, and the things what the user sees in the virtual space as we defined in […].

*Communication*, *operation* and *actor:* The user *communicates* with the *actors*, and *operates* on the *objects* and on the *virtual space*, as we defined in […]. In this paper we go into details mainly about the *operations*.

*Comprehension Assistant*: it is the hypothetic service dialog interface of the IConS technology. It can be imagined as the descendant of the context sensitive help: e.g. for the right click on an *object* shows the object's meta-information. In the other hand, it performs also the setup, configuration and audit dialogs of the portal.

---------

Usability (in recent times we can call it as Information Architecture) is the new and growing area of the ICT technology, namely of the information building for interactive use. Our Reference Model, in short the URM, is basically intended to elaborate a *map of notions*, belonging to this area; which can be the base to elaborate the ontology of the HCI, i.e. the ontology of the Information Architecture discipline.

Nowadays we are over-packed with CMS software technologies, developed by proprietary and open software industry, and used/distributed by system integrators. But, none of them can be thought of as a technically thought-out, theoretically established, exact technology.

In the other hand, the usability discipline by now has no strict conceptual basis – it has no proper set of terminologies and categories to tackle the essence of the usability. The great brains of the usability, Nielsen [1], Krug [2], Shneiderman [3] gave practical methods to evaluate the portals, and, as Brinck [4], to design usable portals. Their focus is basically the e-business, their approach is essentially heuristic and not cover the whole area of the usability.

The ISO standards are hardly referenced in the publications, their impact is low, and they are supported neither by consultants, nor by tools. Bevan [5] gives an introduction into their nature. Moreover, they mainly deal with the low level, the ergonomic part of the matter described in the URM 1st and 2nd levels, see here later.

Our analysis is for product quality, not for the design process. Moreover, our approach is essentially differs from the recent CMS practice, so it establishes a new CMS philosophy, we refer it URM philosophy.

It is very instructive, that dealing with the elaboration of the nomenclature for the CHI, it appeared step by step that it ought to be established either on some kind of the existing CMS-s or some kind of the existing portals, with appropriate level of the functional variety and minimal conceptual strictness. If we found appropriate kind of them, this paper would address how to correct, to upgrade or to extend them. But it came to light neither of them exist or can be available. Therefore we refer the hypothetic - or planned, desired – CMS technology named IConS.

The URM is a heuristic classification of the concepts of the *portals* resulted by a compromise between the human cognitions and the software technology requirements.

The purpose is to elaborate a map of notions, to promote the research of the nature of the usability, and to be able to discuss the *requirements of usability* of ICT services. *It is high time to make the usability a scientific area with strict conceptual basis*. Further goals and possibilities are to build

- a *portal* audit methodology ,
- a *portal* design methodology and
- a *portal* building technologies

complying with the requirements discussed in URM terminology.

We don't believe that by experiments on user's behavior (eye tracking, menu creating by card sorting, etc.) can tackle the essence of the usability as a whole. These experiments, naturally, do yield in important partial results. But the usability is so complex quality of a *portal* that, maybe, we never will be able to put it on strict **experimental** bases. The thing we can argue for that the *portal* as a technical construction must have strict **conceptual** bases. And consequently so must have the CMS software technology.

So our approach is to find out an ad hoc model seeming to be useful, then to build a strict nomenclature of the HCI upon it.

## II BACKGROUNDS

There are only a few trials to analyse the structure of the HCI. Here we refer only Garett [6]. Our analysis is formally similar to that, which splits the HCI to 5 layers, from the concrete *surface* to the abstract *strategy*, and splits the layers into *functionality oriented* and *information oriented* elements according to the basic duality of the user experience, as shown in the Table 1.

| | functionality oriented | information oriented |
|---|---|---|
| Surface | Sensory design | |
| Skeleton | Interface Design | Navigation Design |
| Structure | Interaction Design | Information Architecture |
| Scope | Functional Specifications | Content Requirements |
| Strategy | Product Objectives | |

Table 1: The structure of the user experience by Garett

Our model is more sophisticated. E.g. the notion if IA in our interpretation is not a subcategory of the usability, but at least equivalent or even may be larger.

**The portal as precise technical construction**

It will be very instructive to have a survey of the scientific disciplines of the ICT have produced software functions which is wide-spread around us as part of the desktop platforms.

*The 3rd generation programming languages, in the 1960s.* ALGOL, C, PASCAL. Note, that the countless languages born later on - some of them called 4th generation language - are not exact construction, or are not wide-spread in platform. Note that the OO-languages as C++ and Java aren't of new idea; they are natural extensions of older languages with OO-functions and libraries.

*The inter-process communication, established by Dyksta, in 1968.* The semaphore, the message and the event flag mechanisms are integrated in all multitask platform.

*The relation database management, established by the Codd model, in 1969.*

*The communication networks, based on OSI model from the 1970s.*

*The cryptography, the digital certifications, the 1970s.* The RSA is from 1976.

*The "windows standard", from the 1970s.* This is the ad hoc name used in this paper because it has no widely-known name. It is an ad hoc construction, containing the

- windows management,
- the menu philosophy,
- the pointing device,
- the low level communication tools as radio buttons, writable fields, etc.

We see that except for the last one, these software functions or disciplines has the next features:

- is standard part of the platforms, either being integrated into a platform, or being a product
- has a product name and responsible support
- has strict mathematical bases
- is de facto technical standard, some are canonized in ISO .

In the author's experience there is no HCI-platform with such qualities; moreover, there is no project to develop solution for that. The software developing actors all over the World work either based on their own ad-hoc HCI solution, or based on the 40 years old "windows standard" of poor abilities, without strict scientific base we sow above.

Remember, that the OO methodology and culture is important part of the programming languages. But this culture is not inherited to the HCI, as we discussed in […].

We have to know, that after all the ad-hoc solutions of the software developing actors are based the "windows standard" also. So will be the new IConS technology, naturally. But the mission of our research is that the IConS technology will be a standard set of software technology layers lying on the "windows standard" and will be used by the actors, developing professional applications, if they will intend or will be ordered to use it.

## III. Our Scope: The World Of The Professionality

Naturally, our goal can't be performed in general way. To precisely state the scope, we define some notions.

In the […] and […] we discuss the difference between the popularity and the professionality. The *popular* sphere consists of what we do in leisure time, without liability. In the *professional* sphere our doings may have financial, legal consequences for our livelihood. For example the social networks are mainly in the popular, while e-governments, e-business are in the professional sphere.

The ICT discipline is wanting in the detailed analysis of the differences between these two types of the behavior of the intended audience. This is not the business of the ICT or the Usability professionals: this would be an interdisciplinary matter of the ICT and the sociology.

### Routinism, fast comprehension and exclusiveness

Consider this 3 basic situation of man-machine interaction.

**Routinism**: Operators, hand-workers, geeks, specialists have to perform a large number of the same, practiced operations with the same software, as fast and perfect as they can. The software is generally certified and embedded. Example: use of the surgery instruments, or technology control.

**Fast comprehension**: Customer, client, visitor and guest go to the portal, use the portal, rarely, occasionally and want to find their way at the portal without any misunderstanding. This is the sphere of *public services*. E-business or e-government portals are generally used in this situation. Example: find the call for application in a government's portal, file applications and maintain them later.

**Exclusiveness**: It is a special kind of the situations mainly of the "Fast comprehension", possibly of the "Routinism": a community purchases a software tool for itself. The members use it; get accustomed to using it and do not feel any comprehension problem. The outsider's claim to use the tool is neglected. This is a kind of the IQ-racism, out of our scope.

NB: The experts of the information sciences, namely system designers, programmers, being experts, are inclined to the IQ-racism. Therefore the scope of usability is out of there interest. This is one of the reasons for many usability problems.

### Our focus requirements: the 3 great lacks

In our approach the basic insufficiencies of the recent HCI-s from the point of view of the usability are as follows:

1. The **conceptual correctness** of the interface, it is the base of the Fast comprehension. Our Reference Model emphasizes this requirement.

2. The **ergonomics** of the interface, it is important for the effective manipulation on the site for both situations, in Routinism and Fast comprehension.

3. The **security feeling** is very important to deal with, but security and usability are often antagonistic. Here we deal with the client's security feeling and not with the technical and organizational security. This latter ensures the secure operations of the networks, servers, providers; their knowledge is canonized ISO or CC standard. This area is invisible and unperceivable for the client.

The main area of the security feeling is the authenticity feeling: the client can entrust his or her data to the portal, for preserve them in an invisible place, in a cloud for interactively manipulate them by invisible actors, and prohibit others from seeing it.

The user's security feeling especially concerning to the HCI is a hardly studied area. Here we accept the hypothesis of the common sense that this feeling comes from different things: in the popular word comes from being accustomed to the ICT, namely to the used *portal*, in the professional world from the conceptual and the operational correctness of the HCI as whole. So the security feeling in the professional word is not antagonistic, and moreover it can be in synergy with the usability in our approach.

While in the popularity the importance is: 1. ergonomics, 2. security feeling, 3. conceptual correctness, in the professionality the importance is reversed: 1. conceptual correctness, 2. security, 3. ergonomics. Therefore the ontology and the security (mainly the user's security feeling) are in the basis of the URM philosophy.

## IV. THE URM'S TOP HIERARCHY

We divide the interactive information systems into *layers*, from the point of view of man-machine interaction. The Table 2 shows the top level hierarchy of HCI in the URM philosophy, i.e. as they may be built on each other *in our mind*.

| | | |
|---|---|---|
| 7. | Conduciveness layer | Is the mission useful for the intended audience? |
| 6. | Process control layer | Is the business or service logic correct? |
| 5. | Synopsis layer | Are the connections of the outer world clear? |
| 4. | Domain semantics layer | Is the logical structure of the domain clear? |
| 3. | ICT semantics layer | Is the logical structure of the portal clear? |
| 2. | Simple objects' layer | Are the simple objects operable correctly? |
| 1. | Perceivability layer | Are the objects in the screen recognizable? |
| 0. | Physical ergonomics layer | How I feel myself using the device? |

Table 2: The top level of the URM model hierarchy

Each *layer* refers to some sort of the functionality – on the one hand to human cognitive function, on the other to the software functions. The main considerations to establish the boundary between *layers* as follows:

A. The layers identify the different areas of *human perception requirements*, the *software technologies* and the service *responsibilities*, and demarcate them from each other.

B. The different software layers, stratified on, are separated from each other. The main separation is between the 3-4 layers, the ICT platform and the application *domain*. Both are split into sub-layers.

C. The requirements and compliancy check list of the layers can be independent from each other.

D. We are intended to establish the requirements in that structured and a strict way that it can be the base of a strict layer based audit and DFU methodology, and a layer based portal building CMS software.

E. The URM-based audit methodology must work from down to up. E.g. if user has problems in the layer 1 in recognizing the objects, this causes problems in using the layer 2 in using them, whether the layer 2 complies or not.

This last point makes the URM similar to Maslow hierarchy of general human needs […]. If the need of a lower level does not comply, the higher level needs may lose their importance.

Note, that the layers don't mean any sequence of the time. The user may recognize a well known logical structure of the 3rd layer, than the wrong-formed logo (in the 1st layer). Moreover: general requirement that the user has to recognize the conduciveness (in the 7th layer) before the complicated structure (in the 3rd layer) of the *portal*.

Emphasis, that this model is not a predictive one, which would be to be verified by experiments. It is a descriptive and definitive model, intended to be used for constructive works. The proof of its usefulness will be the successful construction of the audit methodology, the DFU methodology, and finally a CMS technology.

### ***Attributes and services of layers***

To describe the layers we will specify the next six sings for all layers:

{1} *Category (attribute)*: domain dependent, if depends of the domain, the specialty, the software deals with. Else is domain independent, that is depends only from ICT domain.
{2} *Technology area (attribute)*: area, the features of the layers depend from.
{3} *Quality requirements (attribute)*: standards, recommendations concerning to the features of the layer. The quality requirements are the places, *where we can articulate, what is the usable portal***.**
{4} *Security feeling requirements (attribute)*: Here we deal with the client's security feeling and not with the secure operations of the networks, servers, providers - for they are invisible for the client. Te main area of the security feeling is the authenticity feeling.
{5} *Responsible roles (attribute)*: they are responsible for the complying of the layer's function with the quality requirements. The roles are discussed in the CMMI[1] and COBIT[2], we don't detail them here. They are:

| | Manager roles | Technology roles |
|---|---|---|
| Provider roles | provider's audit sponsor<br>help desk (mail, phone) | CMS platform<br>business analyst<br>system analyst<br>ontology expert<br>Internet expert<br>programmer<br>portal editor team |
| User roles | client (user)<br>client's business<br>client's audit | interior (office) design<br>HW supplier<br>client's maintenance<br>desktop platform |

[1] Capability Maturity Model Integration
[2] Control Objectives for Information and Related Technology

Note, that the programmer role has less and less importance in the *portal* technologies. The URM philosophy doesn't need it at all.

{6} *Comprehension Assistant (service)*: It performs the service communication of the layers. Here we mean the communications, object or tools of those are not permanently at the screen. E.g. the client – more precisely: the appropriate role of the clients - can communicate with the layer to manage – to see or to modify – the layer's attributes. The notion *service* means, that it is standard, it is not constructed again and again for all *portals*.

## V. The Conceptual Hierarchy of Entities What We See

Before going in the details about the model, we have to make a demo nomenclature of the set of the entities, we see on the screen, and the *operations* we use to modify them. Naturally it is not complete. It is to demonstrate, how to build up the HCI-ontology in the URM, what kind of analysis is needed for it.

A) *Objects:* is the set of entities of the domain – either *simple* or *compound object*, e.g. the single writable field or a form.

a) *Meta-information* of the *object*, e.g. the creation time, the owner, the authorization (access right). Generally they are not in the screen in direct way to avoid it to be jam-packed. The *meta-information* constitutes the *virtual space*.

i) *Attributes*, *annotations* (unary relations): They are peaces of *meta-information* concerned to only one object. Detailed below.

ii) *Connections* (binary relations): They are the most important pieces of *meta-information* for constituting *virtual space.* Detailed at the URM 3rd layer.

b) *Object management tools* to modify the content of the *objects*. To write in a *writable field* is a trivial example. Other examples the double click to the .DOC icon to call the editor. Editor is one of the *office tools*. However, there are many object management functions not integrated into the HCI, mentioned as 'CarvedScuds'[3] tools: **c**reate, **a**nnotate, **r**ead, **v**erify, **e**ncrypt, **d**ecrypt, **s**ign, **c**ompare, **u**pdate, **d**elete and **s**ynchronize.

c) Tools to modify the lay-out of the *objects*. E.g.

i) to adjust the window size and position containing the object,

ii) to *shift* or *scroll* the *object*s' position beyond the windows,

iii) to diminish or magnify the *object*, adjust the color, the font type, etc.,

iv) to decide, which meta-information are to be displayed.

d) Tools to modify the position of an *object* in the virtual space. E.g. move an *object* from a directory to another.

e) Annotation tools to modify the *meta-information.*

*B) The virtual space*: the connection between the object, expressed by the meta-information. The most well-known *virtual space* is the tree-like directory structure.

a) *Meta-information* (attributes, annotations, type) of the *virtual space* (generally of a part of it, called *sub-space*[4]), e.g. the creation time, the owner, the authorization data.

b) Elements of the *visualization* of the connection of the objects, i.e. the visualization of the *virtual space.*

i) *Boundary***:** may appear to border the *objects*: to separate visually from one another, and shows the togetherness of the parts of the *compound objects*. It is a requirement of the URM 1st layer that the *boundary* must belong to the *object* instead of the *background*, because the *objects* can appear in front of different *backgrounds*.

c) Tools to modify the *virtual space*, e.g. to rename a directory.

d) Tools to modify the *structure* of the *virtual space*, e.g. to create, delete or move a directory.

e) Tools to manipulate a group of the *object*, e.g. to group them to or ungroup them from a directory.

f) Tools to reconstitute (essentially change the structure of) the *virtual space*. These are wholly lacking in the word of the Internet. This paper suggests tools for this.

g) Tools to *share* and *unshared* a *sub-space*,

h) Tools of algebraic operations between *virtual (sub-) spaces*: comparing them, taking the difference, the sum etc. of them.

C) *Browsing tools*. Browsing means the changing the position of the user's focus. It must be able to fix at any *object*. The different input devices e.g. the mouse and the keyboard having different focusing philosophy may have different focuses at the same moment. This effect can be confusing for the users without designer's care.

*Browsing* in some special cases can effect to the *positioning* of the *objects* also, and vice verse. E.g. to navigate out of the screen, may cause the *object* to move beyond the window, see the A)-c)-i) and -ii) in this hierarchy.

D) *Portal setup tools:* modifying the layout or the behavior of the portal as a whole. (The tools, concerning an individual object, or the point of the virtual space, are in the previous (A)-c), A)-e) points.)

E) *Desktop setup tool:* Here are only the things, to set up our computer, so these influence all the portals. Unfortunately in our days one parts of this tools is in browsers, other parts of it are in the portal.

F) *Actors* (users) and *activities*. These important entities will be discussed in subsequent papers.

G) *Ads*. Some sponsors want to prohibit them in the professional *portals,* especially in the government's *portals*, but in the

[3] Ad hoc acronym

[4] The term *domain* in the URM is reserved to the area (specialty) of the application.

URM philosophy we think, it is a need. E.g. a portal can promote new functions of itself.

We have to give some help for this strong and difficult-to-comprehend conceptual hierarchy. Although this hierarchy mainly is for the elaborate the nomenclature of the HCI, it implicitly expresses the authors' opinion of what a *portal* or a CMS is to be like. This symptom can't be avoided when creating such a synthesis.

The entities *object* and *virtual space* are in the top, and the tools to manipulate them are theirs descendant. It comes from the URM's principle, that the *space and objects are primary entities compared with the manipulation*. At first we recognize the word around us then we act[5]. (Naturally, there can be situation, when the user is familiar with the environment, and works by heart with closed eyes. Mostly it is would not be our case of the *fast comprehension* but the case of the *routinism*. The CMS has to work in according to the majority of our cases.)

The CMS has to personalize a portal. E.g. using the e-services every client sees his own folders beyond the things concerning to every clients. Or in an e-shop the user sees the personal shopping basket. There are more complicated situations, when we use a common workplace, e.g. a networked project management tool. The owner and the authorization of an *object* or the *virtual space* – or of a sub-space – are very important meta-information.

About the *virtual spaces*: in the URM philosophy an object must be in a well defined (and at the same time only one) point of the *virtual space*. So it can't exist alone. Therefore we don't send *objects* (documents) for the other *actors* (e.g. colleagues), we share the *sub-pace*, containing the *object* and sent the ULR or DOI instead. Sharing/un-sharing the *sub-space* is analogous to the mount/un-mount of the file systems.

We emphasis that the *objects* are on the screen[6] and the *virtual space* is not on the screen, it is in our mind built upon the *meta-information* we see on the screen.

The *virtual space* has two main functions in recent state of the research: to hold the object in a static way, and assign the workflow to the user and/or other *actor*s.

About the tools: naturally they – mainly the browsing tools - often are not on the screen as widgets, they operate directly by hitting or positioning devices.

A)-a)
i) *Attributes*, *annotations* (unary relations) of the *object*s. We are compelled to define and in some measure "christen" these well known notions, for the subsequent conceptual correctness. There will be some new attributes also, e.g. **seen** or **not-seen**.

(1) **Text-like** or **picture-like** object. Object is *text-like*, if we recognize it at a glance as text. It is difficult to define exactly the differences: to recognize a not European character as a letter is uncertain for European users. One technically strict and useful definition may be: text is the thing which consists of letters having alphabetic ordering. It is not evident whether the captcha is text or isn't.

(2) **Page type** or **office type** (-**icon view** or -**subscreen view**) object.
(a) **Page type**: special object, forming a paper-like sheet at the screen, as it is usual at the web. Each page must be pointed from and accessed by the *table of contents*. Other access possibilities may confuse the user, so it is not correct at the URM's 2nd level.
(b) **Office type, icon view**: we know it as the file type icon. It is connected to call the tool to modify, or manage it.
(c) **Office type, sub-screen view:** differs from the icon view in that it appears immediately with the containing page, as if its icon were clicked.

(3) **Simple** or **compound** object
(a) ***Simple object*** is everything we can see in the screen, having own individual URL (and, as it is increasingly required, DOI, [doi]) and having no internal structure in visual term. A text without header and clickable word, no matter how long is a *simple object*. URM 2nd level requires, the all object have ULR or URI.
(b) **Compound object**: fixed set of (simple or compound) objects, belonging or seeming to belong together. E.g. the 'search' panel consists at least of a 'writable' object, and a 'command' object to search. The main requirement at the URM's 3rd level is what belong together, must seem to belong together, and vice verse.

(4) **Static, external link, internal link, command, dynamic** or **writable** *simple object*. These 6 values exclude each others.
(c) **Static**: its content fixed by portal editor via CMS. Practically it is HTML text.
(d) **External link,**
(e) **Internal ink** is one kind of clickable objects used to browse, to navigate in the *virtual space*. The *internal link* belongs to the *C) Browsing tools*. The URM 2nd level requires that the *external* and the *internal links* differ for at a glance, as e.g. the Wikipedia does it. *Internal links* **must be** element of the *contents* table by the requirements in URM 3rd level. "Straying" *internal links* are not allowed.
(f) **Command** is a clickable *simple object* doing something, e.g. send a request (e.g. the prepared filing). It mustn't cause side effects according to the URM 2nd level: does not navigate anywhere as tools in C), has no effect to the *virtual space* as the tools in the B)-b) or B)-c) have.

[5] This is nor a trivial principle. There are environments or working habits with the opposite philosophy: when at first I start the *office tool* (Editor), and then, from the editor I search the *object* (document) I operate on.
[6] Naturally, as the result of the high quality OO programming the *objects* of the HCI are in the computer as well.

(g) **Dynamic** changes its content independently from the user activity concerned directly to it, e.g. the clock, or the number in the shopping basket showing the sum to be played.

(h) **Writable** is practically a field to type in.

(5) **Document-like** or **panel-like** *object*

(6) **Seen** or **not-seen**: The *object* is seen, if the user has seen it since last modification.

(7) **Annotation** (Modifiable *meta-information*). The pervious meta-information are fixed by the portal designer or editor, or changes automatically. The next one are modifiable in user time. Modification tools are declared in the A)-e) paragraph.

(8) **Workflow-attributes. Filed/not-filed, accepted/rejected,** etc. These are useful attributes for the portals working with document being common for more *agents* (users).

(9) **Authorizations** They are the greater part of the *annotations* describing access right to the *objects*, to enable/disable 'CarvedScuds' tools of the A)-b) paragraph for a user.

There are some entities, connected to the objects, important to recognize them, but not belonging to previous categories:

(10) **Background** (canvas): It is not functional entity. It can carry design which is very important, non-functional, marketing entity of a portal. No we think of it to be a whole for the portal having one layer. The overlapping objects covering up the others (they are mostly the pop-up and the pull down menus) may have own background.

(11) **Design**, or lay out is difficult to deal with, because it is not a independent entity, it can be manifested by other entities as:

b) *Background*. It can be hoax: seeming to the functional entity

c) *Boundary* or other element of the visualization, e.g. he visual arrangement of the functional entities.

## VI. The Description Of The Layers

### *0. Physical ergonomics' layer*

It deals with the ergonomics, and productional psychology. Relevant subjects are the display terminals, screen colors, refresh rate, room illumination, height and angle of the table, of the screen, of the keyboard, mouse and keyboard ergonomics, etc.

{0-1} *Category*: Domain independent.

{0-2} *Technology area*: IT desktop hardware.

{0-3} *Quality requirements*: standards for office ergonomics, e.g. ISO 9241.

{0-4} *Security feeling requirements*: standards for office security. The *security* requirements are described in the chapters of physical security in the security management standards. Unfortunately, they study the security not the *security feeling*.

{0-5} *Responsible roles*: interior designer, HW supplier.

{0-6} *Service dialog*: none, this layer is not in the competency of ICT.

### *1. Perceivability (readability, audibility) layer*

The user recognizes the objects and theirs main attributes. This layer deals with the readability, audibility, of *simple objects*. The accessibility techniques work at this layer.

{1-1} *Category*: Domain independent.

{1-2} *Technology area:* desktop platform: operating systems, browsers, mail clients.

{1-3} *Quality requirements*: Most of these are in the standards for portal ergonomics, canonized in WCAG 2.0 of W3C. Some examples of them:

- Resolution independency: information fits into the screen.
- Color independency: there is no information only conveyed by colors.
- Correct localizations: language codes, number formats, datum formats are readable.
- Screen management: text wrap, window scaling-positioning-overlapping, shifting-scrolling bars, data collisions, data visibility are managed.
- Blinking problems: migraine and epilepsy avoiding
- *Focuses* of all input devices (keyboard, mouse, etc.) are perceivable.
- Font type problems: avoid some type of serif fonts in the raster display.
- Color collisions: don't use green letters with red background.
- Etc.

{1-4} *Security feeling requirements*: any beyond the correct working of the layer, e.g. without accidental flashing.

{1-5} *Responsible roles*: In the URM philosophy it is the desktop platform of the {1-2} 'technology area' paragraph. Namely, if it complied with the URM-philosophy, the provider's technology roles couldn't be able to create *portals*, which do not fit to the {1-3} 'quality requirements' paragraph. Unfortunately, the recent platforms have no mission to do this. Instead, there are software tools – e.g. the W3C checkers […] - for checking the compliance usually from the HTML code of the *portal*. So nowadays the responsible role is the client's audit, executing afterwards a home page evaluation, using the check software.

{1-6} *Comprehension Assistant*: The user is able to set up the above features. E.g. screen setup of desktop systems, the character coding in the browsers, etc. Recently these functions are deficient, occasional, scattered around the tools.

### *2. Simple Objects' layer*

Here are the functions connected with the recognizing the *simple objects* and using them. The W3C […] calls the similar layer of his checking tools as Navigability, or Operability layer.

The special problems of the *office objects* and the *office tools* (e.g. the editor) we discuss in this layer, for the *office objects* are simple objects from the point of view of the *portal*. They generally are an independent *object* somewhere in the *virtual space* and not part of a *page*. (In the 40 years term it is a "file".) The *office tool* has its own window, *virtual space* and *objects* in the window, being not *objects* and *space* of the calling *portal*. The usability of the *office tools*, the conciliation its HCI to the portal's HCI has deep influence to the usability of the *portal*. This type of the conciliation probably can't be performed in with the present commerce windows tools.

This throws light on a greater problem: the *office tools* have developed independently from the portals, and without any usability control. There are studies, that how to build a usable *portal*, but this discipline has no effect to the *office tool* design. Moreover, we investigate, what is the *portal* in effect, but none of us ask what the *office objects* (the documents) are in effect. The usability of the *office tools* must begins with the appropriate definition of the *office objects*. It is desirable and expectable the appearance of an *office tool* set with limited functionality, which can be integrated into the platforms. Beginning example of this trend may be the well known on line document store […]. But recently it is a wrong example because the lack of the possible integration to other portals.

The difficulties increase considering the *forms*. They are to be thought as a special *office object*. The form management tool must be special *office tools.* Recently there is no proper and widespread form management tool set, even if not regarding any usability question.

The professional form management consists of an *office tool* (used by the *portal* editor user) creating a form-model and a form-filling *office tool* (used by different *portal* users) creating the filled *forms* from the form-model. Then it is needed an API for the use of the form data from a software application, and then a strict version management mechanism, to ensure the consistency after the changing the form-model.

The complexity of a high quality form management is not less than that of a database management. It will be the great unsolved problem of the next years, for these difficulties are comprehended neither by the system analysts, nor by the sponsors. The problem of the forms will be difficulty within the difficulties.

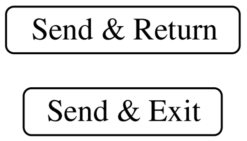
Naturally, in the other hand, the little simple *form*s, as a login form with its two *writable field*s and a *command* are easily integrated in any platform. But theirs HCI is not unified, and differs from that of the great *form tools*.

The cryptography, the digital signature and the certification management are integrated the most platforms, but are not bound to the HCI in usable way.

{2-1} *Category*: Domain independent.
{2-2} *Technology area*: the low layers of the commerce CMS software technology. Recently there is no CMS with such layer.

{2-3} *Quality requirements*:

- General requirement in this layer that the entities built up in the conceptual hierarchy A) – F) must be easily recognized.
- General requirement in this layer the so called "beacon principle": before clicking the user wants to know what will be caused by the click. The bubble text under the cursor position is a proper solution, the (context sensitive) help as we usually face it isn't. Another corollary of the "beacon principle" that the layer has to periodically control the *external links*, and to mark the dead ones.
- *Worldwide identification* of the *objects* is a need by URL, or DOI, and an easy-to-read and telling name. Be the version number the readable part of the name. These are existing functionalities in some applications, but are not integrated into any platform.
- The use of the *office tool* is to be conciliated with the *portal*.
- Static texts have to comply with the *Internet style* requirements […]
- The 'CarvedScuds' functions - of A)-b) paragraph of the conceptual hierarchy - work correctly for the *simple objects*, in a measure that application's semantics in the URM 4th layer needs and the *authorizations* enable them.
- Example for the *commands***:** They mustn't have *browsing* (navigation) side effect, because it could confuse the user. Practically the user may have to get to another *page*, if the actual *page* becomes unnecessary or closed by the effect of the command. It is a requirement, that in this situation the command button unambiguously *declare* this, as in the Picture 1:



Picture 1: A declared side effect

{2-4} *Security feeling requirements:* This is one of the most important layers for this area: has to set up in easy, correct and controllable way the authorization of his or her *objects*.

{2-5} *Responsible roles*: At the URM philosophy it is the IConS technology. Nowadays there is no responsible role. Some kind of the problems can be discovered by the client's usability audit.

{2-6} *Comprehension Assistant*: The annotating tasks for the *simple objects* must be performed at this layer.

***3. Informatics semantics' layer***

The user has to recognize the *compound objects*, the relation of the *objects* to other *objects*, i.e. the structure of the whole *portal,* i.e. the *virtual space*. The portal structure must be coded mainly at the *contents (table)* which is the most important and generally the greatest *compound object*. So the confusion of the *links* and the *contents (table)* is the basic reason for the poor usability of the *portals*.

The name of this layer expresses that the user gets to know, what is the meaning (the semantics) of the *objects*. This is the application independent (with other words domain independent) meaning built up from notion of informatics, demarcated from the 4th level which uses the domain's notions.

***The virtual spaces and the compound objects***

The *attributes* and the *annotations* (unary relations, see A)-a)-i)) mainly used for the basis of the *compound searches*: when we tell the type of objects to search in: it is 'text', or 'caption', or 'comments', etc., we search according to some unary relation. These relations could be the base of faceted search also, but no platform that exploits that.

The *connections* (binary relations, see A)-a)-ii)):

(1) There are asymmetric relations, of which the inverse relation is useful. Here are examples of the type of relations, the use of which is general in some applications. The URM philosophy suggests using them as standards in our platforms in the 3rd level.

  (a) cause_of, consequence_of
  (b) precedence_of, subsequence_of
  (c) explanation_of, application_of
  (d) abstract_of, full_text_of
  (e) etc.

(2) The next relations are the general relation of the 'togetherness' without further semantics. In recent portal practices the *internal link* means this relation and the inverse of it (where are links to this *object*) is not used.

  (a) consists_of, element_of : between a compound object and its element.
  (b) contains_of, part_of: it is similar to the (a). It is between a compound object and group of its element. The group generally is the part of the compound objects we actually see at the screen.

(3) The URM philosophy doesn't allow the "straying" *internal links*[7], but something being like the "shortcut" is really necessary. Therefore we need the next relation:

  (a) lo_contents (Lift out of Content table) It is a special relation between an *internal link* and the *contents.* The URM 2nd level requires that this type of *internal links* is easy to recognize and to distinguish them from the *contents* by their lay-out.

***Some standard compound objects***

The 3rd layer deals with the compound objects. The next definitions detail the A)-a)-i)-(3)-(b) paragraph of the conceptual hierarchy, they naturally are only examples for demonstrate the URM philosophy:

A)-a)-i)-(3)-(b) **Compound objects:**

❖ **set**: unordered set of objects having similar or identical looks and size. It is rare at the screen, e.g. the ornamental or casual arrange of thumbnails. NB: the pull down menu of things for the selection is an ordered one, because the sequence of the things in the menu is an ordering. So it is not a set, it is a list.

❖ **search panel**: the well known panel, consisting at least of a *writable* field object and a *command* objects.

❖ **login panel**: the well known *object*. The simplest one has two simple *writable* fields and a login *command.*

❖ **form panels**: the difficulties of the professional form management were discussed in the USR 2nd level. Here we mean the little forms, being the part of a *page*.

❖ **list**, **table**: set ordered in 1, 2 dimensions.

❖ **Tree** and others: the CMS practitioners and the software developers use various structures of data in the Internet word.

❖ **Contents**: (table of contents) Tree of *internal links*, pointing *pages* and other *objects* the content's editor wants to be registered in it. It is requirement of URM 3rd layer, that each *page* of the *portal* must be pointed from the *contents*.

❖ **Breadcrumb**: sequence of *internal links*, showing the path form the beginning of the portal to the recent page.

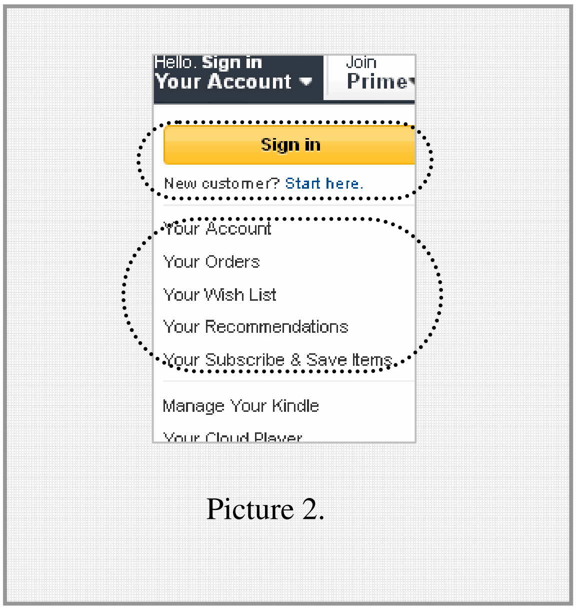
❖ **Portal**: The highest notion of *compound objects* the URM deals with. We also know it as **home page**, **content**, or **site**. Other synonym used the URM terminology: **e-content**. As a *portal* is intended to perform service, synonym also is the **e-service**. It is requirement of URM

[7] This principle is the consequence of the 2. axioms of Object Permanency Principle: an *object* must be at one and only one place at the same time. [OPP]

3rd layer, that a *portal* must have one and only one *contents*.

- **Work area**: is the a subset of the portal, a subspace, where the user can use the 'CarvedScuds' tools declared in the A)-b) paragraph, and the tools on the *virtual space* in the B)-c)…e) paragraphs. *Work area* as a matter of fact is a little virtual desk or virtual office where the user can manage his documents, notes, addresses, check times, etc. connected to the doings performed on the *portal*. See later, at the URM 3rd level.

- **Calendar**: the great suppliers have calendar services with scheduling and sharing possibilities. But the sharing is not integrating into the HCI. The integrating would mean that e.g. a search from some part of my *virtual space,* namely from my *work area* can navigate into the Calendar or can assign the Calendar's address into the finding list.



Picture 2.

For example, see the Picture 2. It is a pull down menu with its ordered elements making up a *list*. The two horizontal lines (hard to see) are *boundaries*, and split the *list* into tree part. The second line in the little dotted area is a *compound object* itself containing a *text-like static* and an *internal link*. The lines in the great dotted area form a *list*. The 'Sign In' is a point of menu also, but at the first glance it seems to be the title of the menu - mainly because of its centered position. The URM 1st layer regards this as (a little) error.

The URM prefers neither the term 'site map', nor the 'top menu'. By the URM philosophy they are redundant notions referring to redundant software elements, so the URM complying *portal* uses neither of them. A well elaborated *contents* has to provide all requirements: allows to recognize and to comprehend the structure (the virtual space), documents the *portal* and makes it referable.

Many of the *portal*-s over the Internet has the redundancy above, they are not URM compliant. So their analysis according to URM is difficult and can only be very rough.

It is high time to equip the *portals* with advanced navigation technologies. The navigation has two well known methods: the browsing and the searching. The advanced sort of them are totally lacking at our interactive environments. Both would need that the *objects* of a *portal* be treated as a little database: have sophisticated set of meta-information and tools for annotation or classification technology to modify/read the meta-information. Both would base on a sophisticated *contents engine*.

There are a lot of classification/annotation techniques. They can be based on different philosophies of the theme map [theme map], the thesaurus [thesaurus], the semantic web [] methodology, and can be implemented via some HCIR methods [HCIR], [Denton] and knowledge visualization techniques []. To refer them and evaluate their usefulness in our research will be later on.

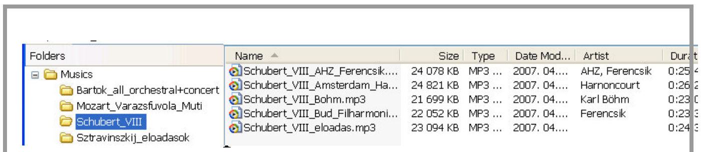
The advanced browsing could base on the dynamic constituting the *virtual space*. In the recent practice, the fixed *contents* – if exists any – visualizes the fixed *virtual space*, because there is only one relation between the *objects*. This only relation nowadays is unnamed; we can imagine it as one sort of 'containing' – if the portal has anything can be regarded as *virtual space*. If the meta-information determine more the one relationship, the possibility to reconstitute the *virtual space* in browse time and consequently the *contents* would very useful.

Remember also, that the mainstream of the research technologies deals with the global searching: they live in the 'low quality portal' and 'high quality global search' ideas: if we are precise and lucky, we will find what we search somewhere in the World. The URM goes the opposite way: we need 'high quality portal' and 'search in those portals exactly we are working with'.

The advanced search techniques would base on the annotation/classification techniques also, e.g. the faceted search [Tunkelang]. We can already face *portals* that have the beginning element of that, the so called *compound search*: when we can tell in the search panel, the type of *objects* to search in: it is 'text', or 'caption', or 'comments', etc. It is interesting, that in the *popularity* the *compound search* is hardly used. The usability practitioners know by experience that it is too complicated for the common people […]. We have to remember here, that the intended audience of our URM research is mainly not the people in the popularity, but that in the professionality instead.

Being advanced or not, all type of searches has two types of results: navigating or creating a matching list. At navigating, the search navigates the user in sequence to every matching point in the *virtual space.* (It must be a *declared side effect* of the *command*.) When creating a matching list, a new the search creates a new object or fills an existing one – it is generally a *page* – with the links of the findings. There is no portal when we can toggle the required possibility.

*Using the work area.*



Folders

- Musics
  - Bartok_all_orchestral+concert
  - Mozart_Varazsfuvola_Muti
  - Schubert_VIII
  - Sztravinszkij_eloadasok

| Name | Size | Type | Date Mod... | Artist | Durat |
|---|---|---|---|---|---|
| Schubert_VIII_AHZ_Ferencsik.... | 24 078 KB | MP3 ... | 2007. 04.... | AHZ, Ferencsik | 0:25 |
| Schubert_VIII_Amsterdam_Ha... | 24 821 KB | MP3 ... | 2007. 04.... | Harnoncourt | 0:26 |
| Schubert_VIII_Bohm.mp3 | 21 699 KB | MP3 ... | 2007. 04.... | Karl Böhm | 0:23 |
| Schubert_VIII_Bud_Filharmoni... | 22 052 KB | MP3 ... | 2007. 04.... | Ferencsik | 0:23 |
| Schubert_VIII_eloadas.mp3 | 23 094 KB | MP3 ... | 2007. 04.... | | 0:24 |

Picture 3. A well known work area.

The archetype of it is in Picture 3. Every more complicated *portal* have such a tool enabling the user to manage their *objects*, but they so much differ from each other both in functionality and in the layout that it is hard to recognize the same essence beyond them. As if the designers thought this type of layout of the operating system as an obscene sights, or they considered the clients too stupid to understand it, or they suffered from their professional spirits to overcome the this poor lay out[8].

The *work area* is one of the functionality of highest level which ought to be integrated into the platforms around us to make them the successor of the 40years *"windows standard"*.

About the *communications* and the *activity theory*.
These are the other functionality of the highest level that must be in the platform, integrated mainly into this layer. This is not detailed in recent paper.

There must be standard *objects* of every *content* annotated as "Imprint", "Owner", "Support", "Mission", "Owner's logo", "Table of contents", "Search panel" etc. Note, that these *object* types were domain dependent about 20 years ago, and because all portals use them by now, we can count them as domain independent. This is the only need which can be satisfied with recent CMS technologies, with an auditable and standard manner.

This layer gives a standard lay-out for the *object* types. E.g. the *work area* type can be like we see at the Picture 3. This feature enables to the developers to demarcate the functionality from the lay-out specified at the URM's 4th level at develop time.

{3-1} *Category*: Domain independent.

{3-2} *Technology area*: the CMS and the content builder methodology. Unfortunately, the commerce CMS technologies are very poor in this. Both data sharing and contents management generally are nowadays a poor and ad-hoc solution, based on the programmer's invention.

{3-3} *Quality requirements*:

- A classification/annotation technology to manage the meta-information,
- Advanced *contents (table) engine* an alphanumeric version,
- The classic searching tool both for resulting correct navigating and matching list,
- Advanced searching and browsing tool,
- W*ork area* management (naturally not the whole functionality; only in the measure that application's semantics in the URM 4th layer needs and the *authorizations* enable them.),
- *Communications* and the *activity* management,
- Standard *objects*.
- Data visualization techniques to display the connections between the objects: it is the graphic version of the *contents engine,*
- *Standard lay-out* for the *objects*.

{3-4} *Security feeling*: both the alphanumeric and graphic visualization are unambiguous. The *Comprehension Assistant* explains all error messages. The encrypt/decrypt/sign functions are simple.

{3-5} *Responsible roles*: Nowadays this layer doesn't exist. In the URM philosophy it would be the IConS technology.

{3-6} *Comprehension Assistant:* Enables access to the invisible *meta-information* which is not accessed by the application. Manages the setup, and reconstitute the *virtual space* for the advanced users.

### *4. Domain semantics' layer*

This is the lowest layer depending on the domain, i.e. the specialty the portal deals with.

The 3rd layer defines the standard classification/annotation data (meta-information) of the objects. This layer adds the domain dependent meta-information. E.g. the *office object* of the type *filled form* in the 3rd layer has the type *tax arrears* in this layer.

The layer enables or disables of the entities defined in the previous layers:

- The object types (e.g. the *work are* type is not necessary for a simple *portal*)
- the *object* management tools (e.g. from the 'CarvedScuds' function only the Read, Compare and the Sign are enough)
- the *virtual space* management functions (these are disable for a simple *portal*)

This layer overwrites the standard lay-out of the *objects* set by the 3rd layer with user friendly and domain dependent lay-out.

{4-1} *Category*: Domain **dependent**.

{4-2} *Technology area*: content builder technology, ontologies. Unfortunately, up to the present we have no

[8] There may be legal reasons also.

portal technology even roughly complying with these requirements.

{4-3} *Quality requirements*:

- Correct use of the content builder functions of the 3rd layer, while building the portal.
- Correct building of the domain independent, in other word ICT, ontology. It can be built once; it belongs to the CMS technology.
- Correct buildings of the domains ontology, it can be built once; it belongs to the portal.

{4-4} *Security feeling*: similar to the 3rd layer.

{4-5} *Responsible roles*: Nowadays this layer doesn't exist. In the URM philosophy it would be the IConS technology and the Provider's technology roles: business analyst, system analyst, ontology expert.

{4-6} *Comprehension Assistant:* Similar to that of the 3rd layer.

### *5. Synopsis layer*

This layer deals with the connection with other portals, the global Internet services, and the physical world.

5.1) Communication with other portals

- ❖ Inter-portal communications (e.g. get data from, put data to the social networks)
- ❖ Service communication about the managements of the common *virtual spaces*.

5.2) Communication with global Internet services

- ❖ Marketing engineering, mail techniques.
- ❖ Domain management: e.g. availability of domain experts, domain bodies.
- ❖ Search engine optimization
- ❖ Global identity management for the *actors* in the low level of authenticity. This practically means the distinguishable and unambiguous assignment of the e-mail addresses and the aliases, as we discussed it at [pargma]. It is the responsibility of the different post office providers in the Internet. The different roles using a *portal* (provider roles, user roles) have to use those e-mail addresses, coming from other providers.
- ❖ Global identity management in high authenticity for *virtual objects*, for individual *actors* and for corporate body *actors*.
- ❖ Connection to the IPR agents for using copyrighted content.

5.3) Connections to the physical world

- ❖ Communication with the (provider's) help desk
- ❖ Authentic printing
- ❖ Authentic scan
- ❖ Inter-modal communication (e.g. Video recording)

{5-1} *Category*: Domain independent.

{5-2} *Technology area*: Various Internet technologies. These functions are generally imported and trusted functions from public portals, post box providers, ICANN, etc.

{5-3} *Quality requirements:* The *porta*l functions must be carefully bound to the Internet services imported/used by the *portal.* There must be on-line guides of their use.

{5-4} *Security feeling requirements:* Various. Its detailed discussion is beyond this paper.

{5-5} *Responsible roles*: provider's audit, help desk, Internet expert.
{5-6} *Comprehension Assistant*: informs the availability of the used services, and the state of the communication with them.

### *6. Process control layer*

This layer contains the business process or working process logic. An information portal, e.g. at the on line media doesn't contain this layer. The required lay-out of the objects is set on the 4th layer.

For providers it is important but neglected doing to have a *documentation strategy*. The documentation strategy ensures that a (user's) role can easily find information for him/her, so that he/she is not made to read through tonnage of texts. The URM has a suggested documentation strategy, not detailed here.

The main point in it is that the *portal* and the *documentation* don't exist independently from each other. The latter is a special linear view of the former, generated by the *Comprehension Assistant*, if necessary – naturally with strict version and authenticity management.

The next topics are to be clearly described and to come out from the screen:

*Business process (or office process) description* – business or office process must be described independently from the IT-technologies, i.e. with user's terminology. The client and all other users must understand their work.

*Office tool declaration* – naturally, the documentation of the used *office tools* can't be integrated into the *portal*. The *portal* has to declare them, and point to their loading URL and their documentation, as the URM 2nd level requires.

*Description of the forms* used in the working process. We mean to describe here:

- The meaning of the fields in the working process
- The control algorithm in the fields, concerning to the process.
- Error messages of the control algorithms.

NB: the *form tool* itself and its description belong to the URM 2nd level. The form independent (i.e. domain independent) error message of the *form tool*, i.e. the messages concerning to the 2nd level operability of the form (e.g. "Obligatory field not filled.") are to be visually differ from messages of the domain dependent algorithms of recent layer. The present-day public and commerce *form tool*s are not sophisticated enough for such niceness.

The possibility of the interruption and the continuation must be elaborated and described.

In this layer the *virtual space* assigns the workflow to the user and/or other *actor*s i.e. guides the users to work.

{6-1} *Category*: domain dependent.
{6-2} *Technology area*: CMS technology
{6-3} *Quality requirements:* This is the only layer in the URM philosophy where the traditional software development, maintenance, audit and quality assurance methodologies have relevance:

Working process must be analyzed, described correctly. The software must be developed in proper quality, etc. All the descriptions have to comply with the *Internet style* requirements.

{6-4} *Security feeling:* the success. The client can fix this doing up at the portal without unexpected difficulties.

{6-5} *Responsible roles*: all the provider roles (except the sponsor), client's audit, client's maintenance.

{6-6} *Comprehension Assistant:* All documentations, descriptions have to be read here.

***7. Conduciveness layer***

The *portal* must have benefit for the society. It must help the intended audience to solve a real problem of their life. The user has to recognize and understand in about half minute the key *objects* of the portal to decide if he/she wants to use the portal or not.

{7-1} *Category*: Domain dependent.
{7-2} *Technology area*: no
{7-3} *Quality requirements:* The Usability practitioner studies […] agree that the key elements commonly are:

- Logo, the identifying the owner
- explanation what serves the portal
- the contents table, the top navigation
- the search panel
- short service mission and/or influence study endorsed by the sponsor

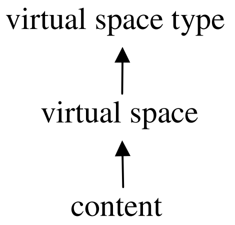
{7-4} *Responsible roles*: sponsor
{7-5} *Security feeling*: The key elements are easy to find, are readable and don't change every day.
{7-6} *Comprehension Assistant*: Helps to find the previous quality elements, which are standard *objects* defined in the 3rd layer. Help to find the documentations described in the 6th layer.

## VII. The Virtual Space Types and Virtual Space Arithmetic

Here we introduce the notion of *virtual space type*.

Let's consider a *virtual space* of tree structure. If some user modifies some objects in the portal, the content of the *portal* will change, and the *virtual space* will remain unchanged. If the user adds or removes objects, the *virtual space* will change also, but the *type* of it, we have called as tree, will be unchanged.

The *virtual space type* of a *portal* is the notion of the higher abstraction we have to characterize a *portal*. Primary we see the physical content, then we try to comprehend the *virtual space* by means of the meta-information in B) paragraph of the hierarchy, finally we understand the type - if any. The levels of the abstraction are in the Picture 4.



Picture 4. The levels of the abstraction in our mind seeing at a portal

If we define the possible *virtual space type* exactly, we can define arithmetical operations between them, mentioned in B)-h) paragraph. This arises when we work on complicated virtual spaces or sub-spaces. The Examples chapter will show a case.

To define the *virtual space type* in an exact way, we can use formal languages notions and tools. The most *portal* has only one fixed *virtual space*. It is rational to assume, that the *virtual space type* of most cases can be described by *regular grammar* if we use the Chomsky's classic typology. The most complicated cases might need *context free* grammar.

## VIII. Examples

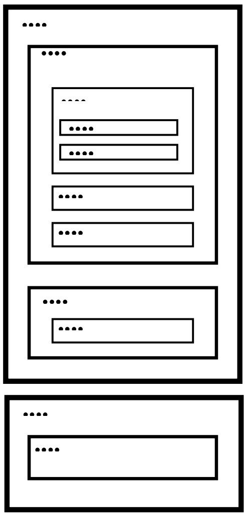
**Virtual space of a forum engine**

At the top of the hierarchy are the 'Topics ', under these users can initiate conversations making a 'Post', and the users can pose 'Comments' to the 'Posts' and to de 'Comments' also.

The Pictures 5-a, 5-b, 5-c show tree different visualization types of the same *virtual space*. The difficulty of the Picture 5-b and 5-c is that they have no proper space for the *contents*. In the beginning when there are a few Topics and Posts the *contents* is unnecessary or can be disturbing, and later, in case of many entities it is required.

Topic1
Post11
+Comment111
Comment112
Comment113
+Post12
Topic2
Post21

Picture 5-a. Browsing-type *contents table*, the most usable, but not the most frequently used visualization.



Picture 5-b. An interesting visualization of a forum. It is difficult to place the *contents table*.

Topic1
.....
Post11
......
Comment111
......
Comment1111
.....
Comment1112
.....
Comment112
.....
Comment113
.....
Post12
....
Comment121
....
Topic2
....
Post21
....

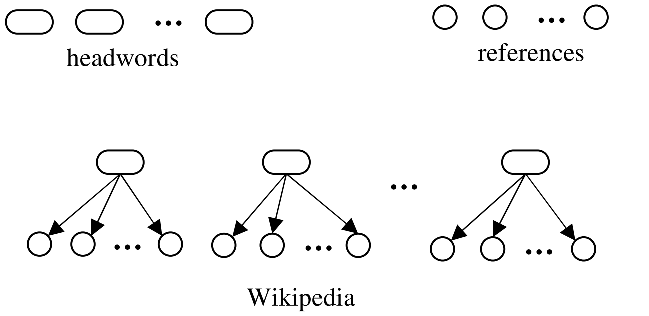
Picture 5-c. The in line, document-type visualization of the forum. It is difficult to place the *contents table*.

**Virtual space of the Wikipedia**

The virtual space of the lexicons is a **set** of object, without relations between them. The alphabetic ordering by the headword can make ordered **list** from the **set**, but we need this only when we search some text through the whole lexicon, as if it were a linear document (similarly to the previous Picture 5-c). The **list** assigns the order of the text search in this case.

Generally we search only by the headword, so the **list** is not visualized in the Wiki.

Other *sub-space* in the Wiki is a simple **set** of references of all headwords. This **set** consists of **internal** and **external links**. The combination of the two *virtual space types* namely the **set** of the headwords and the **set** of the references for all headwords is shown in the Picture 6.



Picture 6. The *virtual space type* of Wikipedia is the result of arithmetic operation between two simpler *virtual space types*.

**What wee see on the Amazon portal**

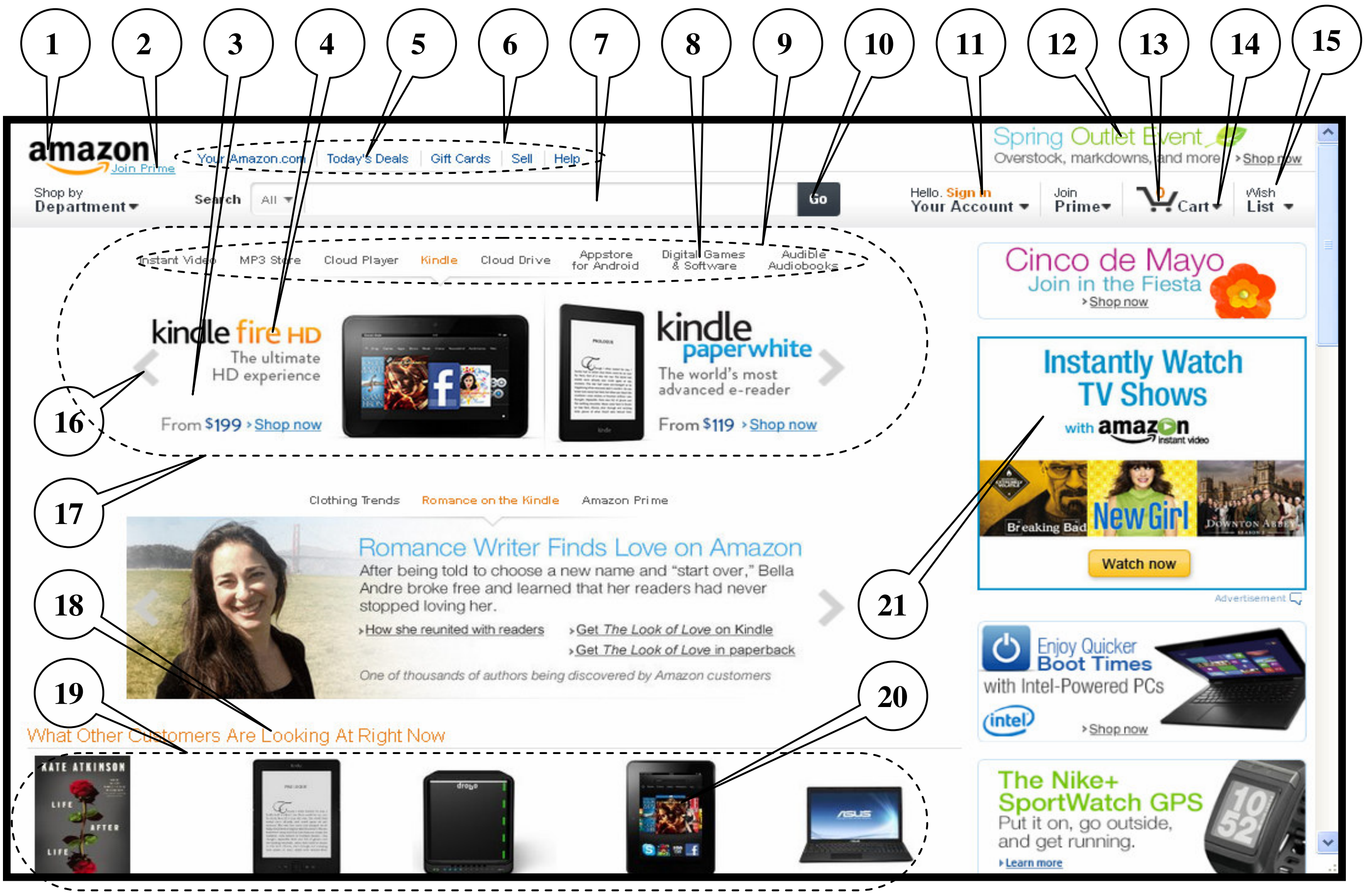
The picture 7. shows the main page of the AMAZON.COM. The Table 3. shows how we recognize the marked entities by the terms of the URM.

| | Level1 | Level2 | Level3 | Level4 |
|---|---|---|---|---|
| 1 | picture_like | static | logo | = |
| 2 | text_like | internal_link | lo_contents | |
| 3 | background | = | eo_compound | |
| 4 | text_like | static | eo_compound | |
| 5 | text_like | internal_link | eo_contents | |
| 6 | text_like | compound | po_contents | |
| 7 | picture_like | writable | po_search | |
| 8 | text_like | internal_link | eo_contents | |
| 9 | text_like | Compound | po_contents | |
| 10 | picture_like | Command | eo_search | |
| 11 | compound | Command | po_login | |
| 12 | compound | internal_link | eo_contents | |
| 13 | picture_like | panel | eo_compound | shopping basket |
| 14 | pull_down | = | = | |

| | | | | |
|---|---|---|---|---|
| 15 | compound | = | lo_contents | |
| 16 | shift | = | = | |
| 17 | compound | = | po_list | |
| 18 | text_like | Static | = | |
| 19 | compound | = | po_list | |
| 20 | picture_like | static | eo_list | |
| 21 | picture_like | static | ad | = |

Table 3. What recognize at the different URM levels in the AMAZON.COM

Abbreviations:
eo_ = element_of
po_ = part_of



Romance Writer Finds Love on Amazon

After being told to choose a new name and "start over," Bella Andre broke free and learned that her readers had never stopped loving her.

›How she reunited with readers

›Get *The Look of Love* on Kindle

›Get *The Look of Love* in paperback

*One of thousands of authors being discovered by Amazon customers*

Picture 7. Objects on the AMAZON.COM (2013.04.20)

## IX. Summary, conclusions

We can draw the conclusions not only from the experiences, but from the theoretical analyses as well.

### *1) Toward the HCI-ontology.*

If we try to build up a nomenclature to deal with the usability problem, i.e. to clear a way to make scientific discipline from the HCI, we face a gigantic construction of conceptual hierarchy can be handled only with great simplifications. Otherwise, we have to see, that the things we wrote in the *'Quality requirements'* attributes can be regarded almost such a degree as functional as qualitative. This throws light on the great problem of the HCI: it was such a neglected area, its business driven development of 40 years yielded in such semantically uncontrolled amount of interactive entities that we have to begin with the beginning, with the redefining the well known notions, in an unusual way.

In this paper about 30 notions of the HCI of the professional office work are defined mainly at the URM 3rd layer. We can estimate that with 2000-3000 notions of the HCI the IConS technology (a new CMS, complying with the principle detailed in this paper) can work in a correct way on the professional application.

The magnitude of the number of notions needed by the ontology of a domain (e.g. the public administration) in the URM 4th level is similar to the former.

Moreover, independently of the ontology the portal must be a mathematically strict construction as well.

### *2) Need of the elaborated synthesis*

We have realized that all the functionalities that are lacking in the HCI of the *portals* and ought to be integrated in the platforms around us are already developed or are under development. Unfortunately these developments are not intended to be a part of the common platforms, but their goal is an independent application or the research.

*The main lack is the integration, the elaborated synthesis.* The fast recognition, the fast comprehension (and orientation) of the complicated things on the screen, the ability to use the Internet as our personal – or in some measure common - electronic office can base on such a synthesis.

### *3) Toward the HCI-standard*

Other conclusion that the quality is mostly based on accepted standards, not only the pure compliancy with a check list. The *portals* have no 'intrinsic' usability degree, the main lacks rise mainly in so far as all portals have different HCI-philosophy. This conclusion is realized in that effort of the URM, that a great amount of the functions must be moved into the platform, and hereby must become some kind of (de facto) standard to ensure some degree of the usability.

### *4) HCI as language: Zamenhof is wanted*

The problem of the usability is similar to the problem of learning and using living languages. Which is the best language? None. The human mind has enough capacity to learn and use anyone of them. But using many languages at the same time is very ineffective and tiring. A HCI-philosophy is like a language. To learn a new portal with the same HCI-philosophy is like to learn new word in the language. It is easy for the human mind. Unfortunately the *portal*s around us speak different, ad hoc and unelaborated and of low quality languages. It is necessary, because none of the portal development project has capacity and competency to elaborate a language of high requirements – i.e. in the URM's terminology the IConS platform.

Zamenhof has elaborated the Esperanto languages based on European languages and the result of linguistics of that time. Now, based on the Information Sciences (ergonomics, semantics, etc.) and the traditions of European office works of many centuries it is possible to elaborate a HCI language.

Linguists say that the evolution of the HCI couldn't be made hurry, for he has its natural speed, as have the human languages. Let's see, the Esperanto is not a successful and widespread language.

The authors don't agree. The whole discussion can't be referred here – recently it is verbal -; here 2 arguments follow:

- The Esperanto had to compete with the settled native languages. Nowadays there is no elaborated construction can be counted as a 'settled CHI language' and would aspirate to be rival of the development the IConS technology.
- The evolution had ten thousands of years to work. The Information Sciences not have, because the unelaborated, insecure, difficult-to-use and business driven technologies will pollute our information environment. Here the responsibility of the scientific community may arise as well.

### *5) Toward the integrated platform*

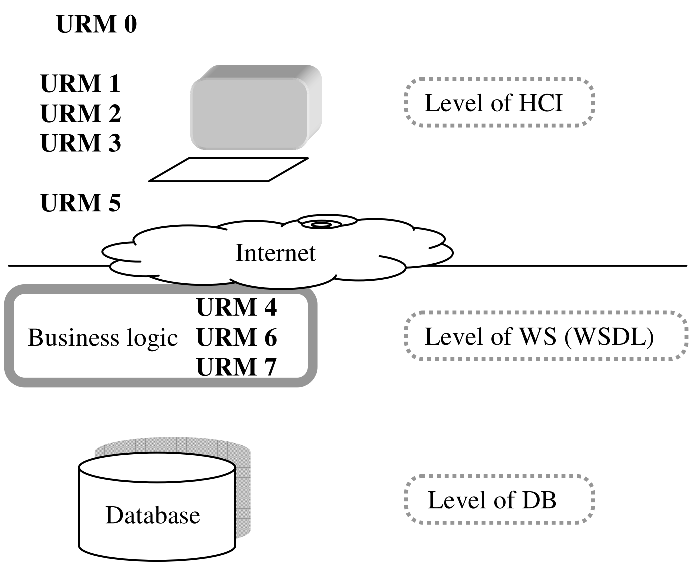
*The Usability problem of the Internet is not the problem of the Internet*: this is the problem of the desktop platforms, inherited to the Internet i.e. to the development of the *portals*. The matter has become significant due to the general use of the *portals* for professional purposes.

Consequently, the term 'integrated platform' and the 'elaborated synthesis' claimed in this paper concern not only the *portal* but our desktop environment also.

The URM's 1st,2nd,3rd levels are independent from *portals*. On the other hand our desktop with my *objects* and *office tools* can be considered as a *portal.*

The picture 8. shows a practical solution. There are database harmonization projects between portals and there are projects to semantically describe a portal in WSDL language. Beyond them our research proposes the HCI harmonization as well.

The URM 1st, 2nd, 3rd, 5th can be work as cloud service, naturally.



Picture 8. The place of the HCI in the harmonization of the e-services

### *6) Toward the final goal: a platform as a product.*

There are so match possibilities for a *portal* - for software as a matter of fact – to be wrong, that there is no method to check correctly its compliance with any checklist or requirement set. Try to imagine, how to check a form management *office tool*, or a *portal* of hundreds of *pages*. Automated check can only work for the URM's 1st and 2nd level – naturally without *office tool* -, as the W3C consistency check does it. The action plan the author's can propose, and that the IConS project is intended to follow is to develop standard technologies:

1. to establish the HCI ontology;
2. to express and consolidate the requirements and the audit methodology of
   - the virtual office for the professional use
   - and/or of the platform and CMS technology do build them.
3. Let the business world to develop platforms and/or CMS technologies complying with our standards. Let the complying platforms compete.

The 1. and the 2. points and the pilot need budgetary investment, – for there is not enough business interest to finance such a research.

It would be very progressive and impressive to use an appropriate subset of a well elaborated virtual reality platform, e.g. the VirCa […]. It would be an escape from the recent Internet environment polluted by the business driven development of the past decades.

Here the appropriate subset is very important, because it must be used from a common SOHO[9] environment.

## Note

Our present analysis is part of a greater research project, intended to elaborate a formal ontology of HCI discipline of the professional use. Based on the ontology, check-lists for portal evaluation and methodologies for the design for usability are being elaborated the professional e-services.

The ontology is anticipated to contain 5-6 basic chapters, covering and philosophically establishing HCI for professional use.

The 1st paper was the [1] on OPP in the HCI.
The 2nd paper was the [2] on the principles on the pragmatics.
This paper is the 3rd one, that clears the way to build up an HCI ontology.
The 4th paper planned will be on Pragmatics and the Activity Theory in the HCI.

## References

[…] there aren't yet correct reference list in this version of the paper.

[Zelevinsky]: Custom Dimensions for Text Corpus Navigation, HCIR Symposium, 2010

[Denton]: How to Make a Faceted Classification and Put It on the Web, http://www.miskatonic.org/library/facet-web-howto.html 2009

[HCIR] Symposium on Human-Computer Interaction and Information Retrieval https://sites.google.com/site/hcirworkshop/, 2007-2013

[Tunkelang] Faceted search http://www.morganclaypool.com/doi/pdf/10.2200/S00190ED1V01Y200904ICR005, 2009

[doi] http://en.wikipedia.org/wiki/Digital_object_identifier

[9] Small Office or Home Office