A Case for *Polyscopic* Structuring of Information

Rolf Guescini[1], Dino Karabeg[2], and Tommy Nordeng[3]

[1] The Department of Linguistics and Scandinavian Studies, University of Oslo
`rolfbg@grace.uio.no`
[2] The Department of Informatics, University of Oslo
`dino@ifi.uio.no`
[3] Cerpus AS, 8342 Alsvg, Norway
`tommy@cerpus.com`

Abstract. We outline the main elements of what we call *polyscopic* structuring of information and argue that information needs to be structured accordingly. The principles of *polyscopy* may both serve as guidelines for creating Topic Maps, and provide orientation for further development of Topic Maps standards and software.

1 Introduction

Information overload is a clear sign that the development of information technology alone will not make us better informed [1]. At the same time, the characteristic problems of today, such as declining health and non-sustainability, impose urgent new demands on the nature and quality of our informing [2]. Paradoxically, while we call our era 'The Age of Information,' our information may well be our stumbling block.

The Topic Maps community is, of course, aware of this problem, better structuring and use of information being its very reason for existence. In this article we argue that in order to be truly effective, the creation and use of Topic Maps, as well of information in general, needs to be oriented according to a collection of principles which we associate with the word *polyscopy*.

Our argument is organized as follows. In the second section, which follows this introduction, we point at a close analogy between the information overload and the crisis in software industry a half-century ago, to which the solution was found in developing the software engineering methodologies, Structured Programming, Object Orientation and others. In the third, we introduce *polyscopy* as an instance of an analogous, methodological approach to creation and use of information, which allows us to instantiate a similar approach to solution. In the fourth, we argue that the change to *polyscopy* is needed if our information should become suitable for its key new role of orientation provider. In the fifth, we show that *polyscopy* in practice leads to a different and, we claim, better way of accessing information, and we illustrate this claim by describing an application where *polyscopically* structured Topic Maps play a key role. In the sixth,

we show how *polyscopy* may be implemented within the current Topic Maps standard and we give some ideas for future development.

Related ideas have been proposed by other researchers (see, for example, [3]). The main novelty in our approach is that we first assess what information should be like in order to best serve its purpose, and then develop a methodology which supports such information. We call this approach *information design*[4], [5].

Polyscopy also allows for *designing* concepts, and thereby giving them a meaning which is more precise and perhaps subtly different from their *traditional* meaning. In this text the *designed* concepts are written in italics.

2   Software Engineering and Information Structuring – A Noteworthy Parallel

A parallel with early history of computer programming suggests the approach to handling the information overload which is proposed in this article.

Early software development projects resulted in thousands of lines of 'spaghetti code,' called so because of their chaotic, spaghetti-like structure, which led the industry into a state of crisis. The solution was found in creating programming methodologies, which provided guidelines for structuring programs [6].

Certain basic principles were shared by all methodologies. Since understanding anything large cannot be done in one piece, abstraction and structuring must be used. The programs need to be structured in terms of small, manageable modules. Underlying this subdivision is the key idea, which provided the basis for abstraction, that programming can be done on different 'levels.' The 'high-level modules' should be constructed and understood in terms of 'high-level concepts,' which are more general and less technical than the 'low-level concepts' which are used in 'low-level modules.' This allows anyone to get an initial understanding of the whole program by reading only the highest-level module which, conveniently, is written in a language which everyone can understand. Similarly, more detailed sub-tasks can be understood by focusing on the particular module where they are handled.

The internal structure of the high-level module should reflect the over-all structure of the program. In that way the high-level module naturally serves as a sort of a structured index for finding more detailed information.

The programming methodologies differ from each other regarding the way in which the modules are supposed to be constructed and put together. Each manner of modular organization reflects a specific way of thinking about programming, which characterizes the methodology and serves as the foundation for its methods. Based on the methodologies, high-level programming languages have been developed which support corresponding modular program organization. The generic GOTO program control instruction, which led to spaghetti code by allowing for arbitrary jumps from one context to another, has been replaced with structured control statements such as IF-THEN-ELSE.

The similarity between the information overload and the 1950s crisis in software industry is quite striking. Now as well as then, a new technology emerged

which allowed us to overproduce, in both volume and complexity. Now as well as then, the routines which were developed for smaller volumes no longer worked for larger ones. Indeed, it is not difficult to see, from the point of view of our parallel, that the hyper-links are similar to the GOTOs, in the sense that they allow for arbitrary jumps from one context to another. The consequence is that the Internet information tends to be structured in a spaghetti-like manner, similar to early computer programs.

In one respect, however, computer programming and information making are different: When a team of programmers can no longer understand the program they are making, the problem is very easy to detect – the program won't run on the computer. Different programming strategies can then be tried until the one that works best is found. When, however, a generation of people can no longer understand the information they have inherited or created, the problem is a lot more difficult to detect, and a lot less comfortable to experiment with.

Fortunately, in handling the information overload we do not need to depend on experimentation. We can learn from history. Our two situations being similar, we can adapt and apply to information the ideas and techniques which have proven to be instrumental for managing the complexity of computer programming.

The first and most important idea is to base information structuring on a methodology.

3   *Polyscopic* Structuring of Information

Polyscopic Modeling has been proposed as a prototype *information design methodology* [4]. In Polyscopic Modeling terminology the *scope* is the point of view, determined in practice by the choice of the subject, terms of the language, epistemological assumptions, methods for establishing facts, representation techniques etc. The *polyscopic* structuring of information, which is one of the main propositions of this methodology, is based on the insight that the *scope* determines the *view* (our way of looking and communicating determines what we are able to see and express). As the practice which follows from the Polyscopic Modeling Methodology, *polyscopy* employs conscious creation of multiple *scopes* and *views.*

The metaphor of the mountain is used as practical guideline for the practice of *polyscopy.* The triangle is used as ideogram to represent it. Every point on the mountain, and on the triangle, represents a viewpoint or *scope.*

Although while taking a walk on the mountain we can see an infinite variety of landscapes and details of natural life, we are not plagued by the information overload. The reason is obvious: The physiology of our vision is such that we always focus on a single, limited *scope* at a time. We can either see an ant, or a tree, or a forest, but never all of them at once. One of the key ideas in *polyscopy* is to preserve this essential property of our vision also in creation and structuring of information.

We say that a *scope* (and the corresponding *view* or piece of information) is *coherent* if it corresponds to a single way of looking, or metaphorically, to a single viewpoint on the mountain. What we see with our eyes is always *coherent.*
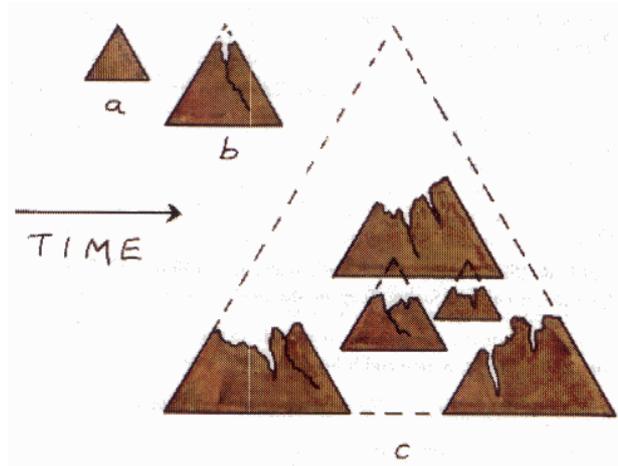
Fig. 1. The Information Fragmenting Ideogram

In *polyscopy* the information is organized into modules in such a way that every module represents a single *coherent view*.

The *scope* just created allows us to understand why we *do* have an overload of man-made information. The Information Fragmenting Ideogram (Fig. 1) suggests that our information is evolving as a collection of 'islands' corresponding to various *informing traditions* (such as molecular biology, sports journalism and Buddhism). Each 'island' is characterized by its own *scope* or collection of *scopes*. If one happens to be living or working on one such 'island,' it is difficult to know what sorts of information exist in other 'islands.' Furthermore, each of the 'islands' is lacking the *high-level* part which would give us an overview of the information it contains, and help us comprehend it. The information overload, suggests this ideogram, is not the result of having too much information, but of having too little *high-level* information which would connect those fragmented pieces together and allow us to make sense of them.

The *polyscopic* information, which is the goal of *polyscopy*, is represented by the Polyscopic Information Ideogram (Fig. 2). The 'i' in the ideogram stands for 'information'. This 'i' is composed as a circle on top of a square. It is suggested that *polyscopic* information consists of two distinct parts: The *high-level* part, represented by the circle, and the *low-level* part, represented by the square.

We have seen that abstraction and modular organization have been the key to managing the complexity of computer programming. Polyscopic Modeling provides three kinds of abstraction for organizing information into modules: *vertical*, *horizontal* and *structural*.

The *vertical* abstraction, which is symbolized by the circle, can easily be understood as, metaphorically, climbing up the mountain in order to see the simple 'big picture.' As suggested by the circle, the *vertical* abstraction involves rounding off the details and presenting the main point.

The *horizontal* abstraction, which is symbolized by the square, can easily be understood as, metaphorically, looking from a side or as projecting the object
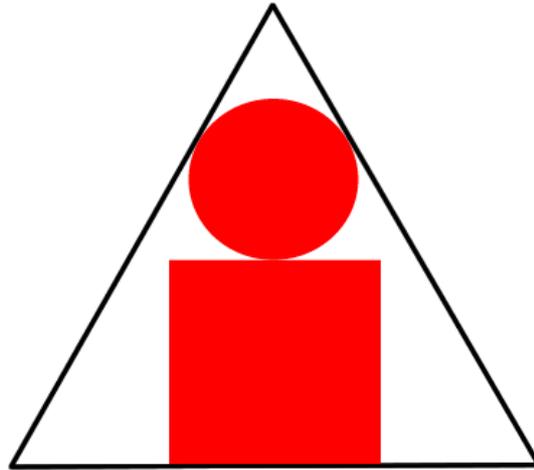
Fig. 2. The Polyscopic Information Ideogram

onto a plane. One again sees a simple picture, but the simplicity is now the result of 'projecting' rather than of 'rounding off.' As suggested by the square, the art of *horizontal* abstraction involves finding a suitable collection of *aspects* (or 'angles of looking') which, like projection planes in technical drawing, give simple and relatively independent (or 'orthogonal') views of the subject, and together give a complete representation of the whole.

The *structural* abstraction, which is symbolized by the triangle, focuses on the relationships which exist among the created *views*. While the *vertical* and the *horizontal* abstraction allow us to create the information modules, the *structural abstraction* allows us to put those modules together.

By inscribing the 'i' in the triangle, it is suggested that the *polyscopic* information is organized based on a structure of *scopes* which the triangle represents. It must be emphasized that the division of information into *levels* does not imply that the modules must be structured as a hierarchy. Indeed, the main point behind the *structural abstraction* is that the structure is an essential property of information, which should neither be removed nor imposed.

Closely related to *structural* abstraction is *polyscopic* navigation. In this context, the principle that the *scope* determines the *view* demands that the user of information be given the capability to choose one out of a number of provided *views* or modules by selecting the corresponding *scope*. As when walking on a mountain one is always aware whether one is looking at a scene from above or from below or from a distance or from close, in *polyscopic* information presentation we support this awareness also in the virtual world. This means that we need to organize the navigation in such a way that the reader has a clear intuitive idea of the *scope*, as if she were walking on the mountain. Visual and other presentation techniques such as animation here have an important role.

The Polyscopic Modeling Methodology provides four criteria to orient the creation and use of information [7]. The one that is most interesting for
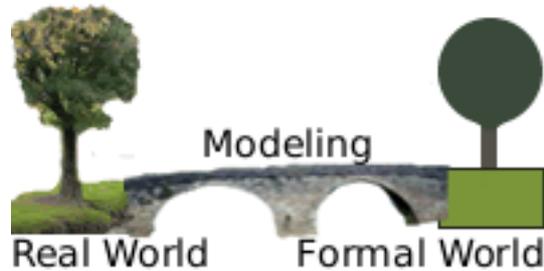
Fig. 3. The *high-level* view of Algorithm Theory course

information structuring is the Perspective Criterion, which postulates that information needs to provide a clear and correct *perspective* (idea of the subject as a whole). Factual truth, which is presently our dominant criterion, directs us to focus on facts and to search for them there where we can find them with largest possible accuracy. This in practice leads to fragmented and *low-level* information. The *perspective* as criterion directs us to seek the information that we are lacking in order to understand the whole, and to present that information in a way which makes the character of the whole and the relevance and the roles of the details clear. In other words, the Perspective Criterion supports the creation and use of *polyscopic* information.

The Polyscopic Modeling Methodology also provides methods for creating *polyscopic* and in particular *high-level* information, as well as prototype examples of such information [5].

For illustration, we now show how an existing body of information, the Algorithm Theory course at the University of Oslo, has been made *polyscopic*. Since this example points at a whole range of problems and possible solutions, in education and elsewhere, we explain it in some detail.

Here is how the algorithm theory has been introduced to the students of the redesigned Algorithm Theory course. What interests us when we study algorithms (and computer science in general) is 'How to solve problems efficiently by using a computer.' But in order to be able to say anything about this question in a precise or academic way, we must first give precise meanings to the real-world notions 'problem' and 'solution,' and define under what conditions a solution may be considered efficient. To that end we create another, 'formal' framework or domain, where only defined or formal concepts exist so that questions can be answered in a rigorous and precise way, and we translate our problem into the language of that domain. Or to use a metaphor (Fig. 3), we take our real-life questions over the modeling bridge into the formal world, we answer them there and we bring the answers back to the real world, where we interpret them and apply the acquired insights in practice.

Contrary to its potential to serve as framework for understanding the main questions about computation in a systematic or academic way, algorithm theory is often considered as an obscure academic pursuit, as 'theory for the sake of theory.' One of the reasons for this misunderstanding is the 'monoscopic' character
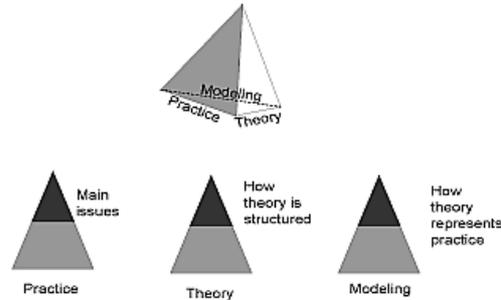
Fig. 4. The structure of Algorithm Theory course

of our information: If one needs to choose a single *scope* to represent algorithm theory, what could be a more natural choice than the algorithm theory itself? The result is that a typical algorithm theory textbook begins with a definition and ends with a theorem. For most computer science students and practitioners, the meaning and purpose of the theorems about 'formal languages' and 'Turing machines' remains rather obscure.

Introducing *polyscopy*, we distinguish three main *aspects* and two main *levels* of algorithm theory (Fig. 4). The 'practice' *aspect* and the 'theory' *aspect* correspond to the two shores of our metaphorical river. The 'modeling' *aspect* corresponds to the bridge.

Standing on the mountain top on the real world side of our bridge, we see only the main practical issues, such as whether there are problems which cannot be solved by algorithms. In the *high-level view* of the modeling *aspect* we examine why the 'formal language' and the 'Turing machine' represent respectively the real-world notions of 'problem' and 'solution.' In the *high-level view* of the theory *aspect* we learn to recognize the main building blocks of the theory.

The students are encouraged to avoid memorizing the theorems and proof techniques and to use the 'mountain-top views' to distinguish and understand the main issues. The *high-level views* provide the motivation for studying the theory, and the context for understanding the details.

After this *polyscopic* re-organization, the course ended up being quite different from what it was before. Notice that this difference reflects not only the way the information is organized and presented, but also the underlying 'philosophy' or approach. In the *traditional* scheme of things, the goal of an algorithm theory course is to teach algorithm theory. In the *designed* course, our goal is to teach what the students most need to know, in a most accessible way.

4   Information as We Need It

As already mentioned, *polyscopy* is founded upon an approach to information creation and use which we are calling *information design*. We now motivate this approach by looking 'from a mountain top' at the way our information presentation developed historically.

The oral tradition, which was the early form of communication, is inherently sequential. Although paper, being two- dimensional, could have been used for implementing a variety of information structures, it was used mainly for making the traditional way of communicating more efficient. In effect, the two- dimensional paper was turned into a one-dimensional sequence of characters and lines, for recording the traditional narratives. Since the printing press also only automated what the traditional scribes were doing, we should then not be surprised if the Internet too has mainly been used for displaying and sharing traditional documents.

The information overload reminds us that such *traditional* way of developing and using information technology (where we simply reproduce in the new technology the sorts of information we have inherited from our ancestors) cannot continue forever. Our *traditional* way of using technology also prevents us from taking real advantage of technology.

Information design is, by definition, an approach to information which is alternative to the *traditional* one. In *information design* the way we create and use information is not automatically inherited but *designed*, aiming to best respond to the needs of contemporary people and society. Information is designed according to state-of-the-art epistemological and methodological insights and by taking advantage of available technology.

Information design is also an initiative to put the *information design* approach to information into practice, which has originated at the University of Oslo.

We can now approach our issue of information structuring from the point of view of *information design* by asking 'What key purposes should information perform in our individual and social lives?' and 'What should information be like in order to best serve those purposes?'

We find that information in our post-traditional, rapidly changing culture needs to fulfill a new role - to allow us to make choices consciously (with awareness of their consequences)[8]. Sustainability is one of the contemporary issues which makes this new role of information vitally important [9]. *Polyscopy* is proposed as a way of creating and using information which suits this new role [10]. By seeing 'from a mountain top' we can comprehend the issues and choose suitable ways of handling them [5]. By consciously seeking to see multiple *aspects*, we can avoid focusing on only one (for example economy) and neglecting others (for example ecology).

5   *Polyscopy* as Orientation for Topic Map Engineering

Every good organization involves an artful combination of freedom and discipline, without which freedom all too easily dissipates into anarchy and chaos. The resolution of the software engineering crisis too involved a change from completely free to a directed structuring of programs. The Object Oriented Methodology, for example, *prescribed* how programs should be organized into modules. The IF-THEN-ELSE statement *restricted* the sequencing control in the interest of clarity. Underlying such prescriptions is an intuitive conception of

programming as it should be in order to lead to well-structured programs. In object orientation, programming is conceived of as modeling the real world in terms of objects and the functions they perform for other objects.

Similarly, *polyscopy* is, above all, a conception of a good way of creating and using information, which is founded upon *information design* as approach, as we have just seen. The *information design* as foundation allows us to determine which of the available ways may be considered as 'good.'

From this vantage point we may distinguish three modes of structuring information:

- Linear. This organization is typical for traditional university courses and textbooks. The interaction mode it supports is passive, sequential assimilation by listening or reading.
- Semantic. This organization is characteristic for Topic Maps as they are today. The interaction mode it supports is browsing, guided by free association.
- Polyscopic and semantic. This organization is the substance of our proposal. The interaction mode it supports is to use the *high-level* information as one would use the view from a mountain top, to acquire a quick understanding of the subject as a whole and its main elements, and to direct further inquiry based on this view, aiming at a correct and clear *perspective*.

While the latter two modes share the advantage of providing the information the user wants, thus also engaging her interest and receptiveness, the third mode has the additional advantage that it supports the holistic understanding of phenomena and issues, which, as we have seen, is our prime socio-cultural need. At the same time, by providing the 'mountain top view,' and organizing the details into *aspects* and modules, the *polyscopic* organization allows us to control the overload.

We illustrate the above ideas by describing a real-life example.

'Flexplearn' is the name of a flexible and exploratory university course model developed by the authors, where *polyscopically* structured Topic Maps play a key role [11]. This model has been implemented and used within the University of Oslo Information Design course.

In the Information Design course the students learn in part by co-designing the Information Design course and the course materials.

If the student should venture into a new field and be allowed to learn by exploring actively and freely, what could be more natural than to provide the guidance in terms of a (topic) map? This is exactly how the things are done within Flexplearn. The Topic Maps are used both for organizing the learning resources, as container for student's results, and for recording the student's 'itinerary' through the information design topics for the purpose of the exam (Fig. 5).

The use of *polyscopic* Topic Maps in Flexplearn supports the corresponding style of learning and understanding. At the beginning of the course the student stands 'on top of a mountain' looking at *information design* as a whole, and seeing the main areas of knowledge it consists of. In order to be able to depart from the habits of the tradition and *design* information, one needs to know about
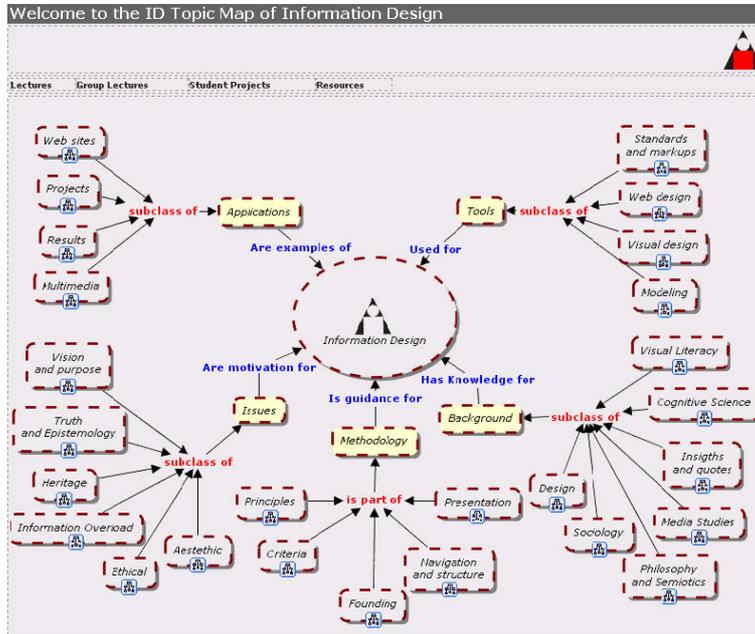
Fig. 5. The Flexplearn taxonomy

the issues which motivate such departure, acquire certain epistemological and other background insights from several traditional disciplines, and be familiar with the technological tools for handling information. From this standpoint, the student can access the information as needed, aiming to gain a functional over-all understanding of *information design.*

The *polyscopic* organization of information also allows us to take care of the prerequisites, which in a traditional course are automatically handled by the fixed order of presentation. By beginning 'from a mountain top' the students naturally acquire an understanding of basic ideas and principles which then serve as background and motivation for more detailed knowledge. While many different learning paths may be taken, each path begins 'from the mountain top' and descends gradually downwards. A simple, hierarchically structured taxonomy (Fig. 5) serves as 'meridians and parallels,' for placing the learning resources and other topics.

6   Implementing *Polyscopic* Topic Maps

As we have seen, many of the constructs described within the polyscopic methodology map readily to Topic Map constructs. To begin with, 'scoping,' one of the strong Topic Maps features, allows us to implement *scopes* in a natural way.

Implementing the *vertical* abstraction should be fairly trivial by using Topic Map constructs. The modules themselves would necessarily have to be

implemented by creating topics for each module, while the vertical structure showing the relations between the modules would call for the creation of association types showing the nature of the structural metaphor used. Examples would be "high-level-view" and "low-level-view", "more-detailed" and "less-detailed", "'part-whole' etc. The existence of the "superclass-subclass" concept, specified in XTM 1.0 for the definition of class hierarchies, shows how associations can be used to express hierarchies and other structures which distinguish levels using Topic Maps.

Example

```
/* TOPIC TYPES */
 [high-level-module = "High Level View"]
 [low-level-module = "Low Level View"]
 [high-level = "High Level"]
 [low-level = "Low Level"]

/* ASSOCIATION TYPES */
 [high-low-level
        = "High Level view" /high-level
        = "Low level view" /low-level]
[topic1:high-level-module = "topic1"]
[topic2:low-level-module = "topic2"]

high-low-level(topic1:high-level, topic2:low-level)
```

The *horizontal* abstraction may be implemented by using the inherent scoping abilities of Topic Maps. Here we have, however, several possibilities. Whether to scope the associations or the occurrences depends on the situation and the kind of presentation or document one is designing. If one wants to say anything specific about the scoped entities themselves, such as relating them to other modules, one would have to create topics reifying them, one for each *aspect*, and relate them to the *high-level* topic directly by association, and then scoping the associations used for creating the vertical relations.

On the other hand, if the presentation is content-centered and does not rely on attaching specific associations to the aspects of the subject, it is preferable to put the scope on the occurrences. One would have to create topics for the modules representing the various levels of information within the presentation, and then scope the different occurrences holding the content itself, or the references to content for a given module.

Scoping Occurrences

```
/* TOPIC TYPES */
 [high-level-module = "High Level View"]
 [low-level-module = "Low Level View"]
```

```
/* ASSOCIATION TYPES */
 [high-low-level
        = "High Level view" /high-level
        = "Low level view" /low-level]


/* ASPECTS * /
 [aspect1 : aspect = "Aspect 1"]
 [aspect2 : aspect ="Aspect 2"]
 [aspect3 : aspect = "Aspect 3"]

/* Instance and scoping example*/
[topic = "The subject in question" @"uri.to.psi"]
[topic-h : high-level = "High level view of topic"]
{topic-h, content, [[Content...]]}

[topic-l : topic
    = "Low level module of topic to represent the low level"]
{topic-l, content, [[Content...]]}/aspect1
{topic-l, content, [[Content...]]}/aspect2
{topic-l, content, [[Content...]]}/aspect3


[topic-l2 : topic = "Low level module relative to topic-l"]
{topic-l2, content, [[Content...]]}/aspect1
{topic-l2, content, [[Content...]]}/aspect2
{topic-l2, content, [[Content...]]}/aspect3

high-low-level(topic-h : high-level, topic-l : low-level)
high-low-level(topic-l : high-level, topic-l2 : low-level)
```

The question remains, however, whether the available Topic Map implementation tools can secure that Topic Map designers, and no less importantly the broader public, will be able to use the Topic Map model to create holistic information as this article envisions it? It may well be the case that in order to develop this possibility to its full potential, high-level constructs will need to be created on top of the existing Topic Map model.

A field study of educational Topic Maps authoring using the TM4L Editor showed that authors generally don't have problems selecting appropriate learning content and resources for their Topic Maps. The difficulties lie, however, in structuring the content by using Topic Map constructs. Topic Map authors are often untrained in information classification and lack controlled vocabularies and support from ontology analysts [12].

Could it be that the creation of some kind of 'Topic Map design pattern' for *polyscopically* designed Topic Maps might help the information designer in her work? By formalizing design insights as a pattern, designers and programmers would be able to talk to each other about them, compare different patterns

for a given solution, and develop a reusable 'tool-box.' [13]. Then tools or even software like TM4L could be built around these patterns, making it easier for the practicing designer to concentrate on creating the *high-level* information that is needed. This would also make it easier for authors to create Topic Map content.

7   Conclusion

In the spirit of *polyscopy*, we condense our discussion to a simple and intuitive *high-level view*, by saying that "**We can come out of the information jungle by climbing to a top of a mountain.**" Like the view from a top of a mountain, *polyscopic* information can give us simplicity and clarity, highlight what is large and important, orient our information search and help us not get lost in the information jungle.

We have also advanced a more general proposal, to base the information structuring on a methodology. *Polyscopy*, or Polyscopic Modeling, is an example or a prototype of this approach. The methodological approach may lead Topic Maps engineering through similar developmental stages as the ones we have witnessed in computer programming (the formulation of methodologies, the development of high-level structuring constructs and standards, development of information design environments akin to programming environments and others). Visual and other media, programming, animation, sound and a variety of presentation tools and techniques will naturally be used to give information multiple 'dimensions' and in that way also suitable structure. Dynamically created structures, based on user profile and needs, will be supported. It may be expected that not only the remedy to information overload, but also a far more positive social and cultural role of information will be the result of this development.

References

1. Postman, Neil: Informing Ourselves to Death. Keynote speech at German Informatics Society conference, Stuttgart, October 11, 1990. http://www.mat.upm.es/ jcm/ postman-informing.html
2. Meadows, Dennis, L. et al.:Limits to Growth. Universe Books, New York, 1972.
3. Bush, Vannevar: As We May Think. In: Atlantic Monthly (1945).
4. Karabeg, Dino: Designing Information Design. Information Design Journal 11 (2003/03) 82-90.
5. Karabeg, Dino: Information Design. Book manuscript.
6. O. Dahl, E. Dijkstra, C. Hoare: Structured Programming. Academic Press, (1972).
7. Karabeg, Dino: *Polyscopic* Modeling Definition. In Griffin at al (Ed.): The Turning of the Tide. Selected Readings of the IVLA, (2004).
8. Peccei, Aurelio: The Human Quality. Pergamon Press (1977).
9. Laszlo, Erwin: The Choice: Evolution or Extinction? A Thinking Person's Guide to Global Issues. Putnam (1994).
10. Karabeg, Dino: Information For Conscious Choice. Information Design Journal 11 (2003/03).

11. Karabeg, Dino, Guescini,Rolf, Nordeng, Tommy: Flexible and Exploratory Learning by Polyscopic Topic Maps. 5th IEEE International Conference on Advanced Learning Technologies, ICALT 2005, July 5-8, 2005, Kaohsiung, Taiwan, 946-948. http://www.win.tue.nl/SW-EL/2005/swel05-icalt05/final/W3-2.pdf
12. Dicheva, Darina, Dichev, Christo: Authoring Educational Topic Maps- Can we make it easier?. Proc. IEEE ICALT (2005) 5th IEEE International Conference on Advanced Learning Technologies, ICALT 2005, July 5-8, (2005), Kaohsiung, Taiwan, 216-219.
13. Ahmed, Kal: Beyond PSIs, Topic Map Design Patterns. Extreme Markup Languages, Montreal, Quebec, August 4-8, (2003).