PART 3

Conclusion & Appendices

CHAPTER **9**

Conclusion

Wrap Up

The dissertation has started with an analysis of the domain of open hypermedia systems, based on an overview of major milestones in hypermedia research, an examination of a requirement list for open hypermedia systems, a scenario for using open hypermedia systems in an engineering environment and the flag taxonomy for open hypermedia systems. Next we have presented our personal perspective on open hypermedia systems based on the notion of a design space and the puppet master metaphor. The former represents an open hypermedia system as a tailorable volume in a three dimensional axes system — storage, presentation and navigation. The latter identifies three levels of tailorability — domain level, system level and configuration level. We have shown that the notion of a design space with three levels of tailorability covers the key requirements for the design of open hypermedia systems, and —more importantly— that it can be generalised for any other open system.

Next, we have investigated object-oriented frameworks and meta-object protocols, two state of the art techniques proposed by the object-oriented software engineering community.
The concept behind an object-oriented framework is that clients reuse the framework design by supplying concrete components for the open ends of the framework. The open ends of the framework are specified using template methods. A framework comes with a number of contracts: sets of rules telling how framework components should be assembled to work properly. Basically, there exist two approaches for specifying reusable designs: the white-box approach based on class inheritance and the black-box approach based on object composition. In general —for the important design decisions in a framework— the black-box approach is favoured over the white-box approach, because object composition allows to reconfigure the framework at run-time.
A meta-object protocol is an abstraction technique where the interface of a module in an object-oriented framework is split in two separate parts: the base level interface and the meta-level interface. The base level interface is used to access the basic functionality of the module, while the meta-level interface allows us to control designated implementation aspects of that module. Experiments have shown that meta-object protocols make systems more adaptable. However, it is recognised that the design of a meta-object protocol remains an open question. First, because it is hard to recognise when something is part of the meta-level interface; second —and this is a fundamental problem—, because it is too expensive to develop meta-level interfaces incrementally. We have coped with the first problem by adopting a meta-level criterion based on the notions of 'about-ness' and 'causal connection'. For the second problem, we have proposed a methodology based on a proper combination of object-oriented frameworks and meta-object protocols.

Our methodology is based on the insight that explicit representations of framework contracts are part of the meta-level, which leads us to the conclusion that *the design of a framework provides an initial ground for the design of a meta-object protocol*. As a consequence we can be confident that the design of the meta-level interface can reuse parts of the design of the framework's base level interface. This is especially important as lots of promising techniques for designing base level interfaces are under way.

The methodology we propose is summarised in four design guidelines

1) Devise a design space for the problem domain.
2) Each design space axis should correspond to a black-box template method.
3) Each design space axis should correspond to a framework contract.
4) The configuration of the design space axes should correspond to a framework contract.

We claim that by following these guidelines, it is possible to build a system for the intended design space that incorporates the three levels of tailorability: domain level, system level and configuration level. As these three levels of tailorability cover key requirements for open systems, our methodology is well suited for the design of open systems in general.

The second half of the dissertation presents an experimental validation of our methodology in the domain of open hypermedia. It provides a description of the so-called Zypher open hypermedia framework as a design pattern language, which is a state of the art technique for describing framework design.

The design space mentioned in the first design guideline is the three dimensional hypermedia design space —storage, presentation, navigation— that resulted from our analysis of the domain of open hypermedia systems.

The initial design of our framework builds on the design guidelines summarised in the flag taxonomy for open hypermedia systems. As such, our design incorporates a separation between storage and presentation on the one hand and a separation between contents and structure on the other hand. Although the flag taxonomy provided us with a good factorisation of the problem domain, it enforces a white-box approach for incorporating the hypermedia design space. To allow run-time reconfiguration, we applied our second design guideline by extending the design with black-box template methods for each of the axes in the design space. This way, we show that *domain level tailorability corresponds with the base level of a framework*.

In a next step, we incorporate system level tailorability in the design of our framework. Following our third design guideline, we see that this corresponds with wrapping additional behaviour around the execution of the framework contract. Favouring a black-box approach, we extend the design to incorporate an object that is an explicit representation of such a framework contract. As we know that an explicit representation of a framework contract is a meta-object, we illustrate that *system level tailorability corresponds with a meta-level of a framework*.

In a final step, we incorporate configuration level tailorability; a process that is very similar to the previous step. Based on our fourth design guideline, we see that this corresponds with changing the execution of the configuration contract and again we extend the design with an explicit representation of that contract. The fundamental difference with the system level tailorability extension, is that the latter extension is part of the meta-level of the meta-level. So we conclude that *configuration level tailorability corresponds with a meta-meta-level of a framework*.

Besides providing the experimental validation for the proposed design methodology, the Zypher framework serves a second purpose: its abstract design was made concrete in a working prototype of a so-called framework browser. A framework browser is a software system that supports software engineers in the management of complex object-oriented architectures, by providing a seamless integration between the design and theimplementation of frameworks. This framework browser has actually been used to manage the relationships between the Zypher design pattern documentation and the source code of the implementation. For example, it is possible to navigate between design pattern documentation displayed using a third-party text editor and the framework implementation displayed using the programming tools that are part of the Smalltalk programming environment. Also, we are able to generate the necessary structures to make this integrated environment accessible over the world-wide web.

The Research Hypotheses Revisited

In the introduction we have formulated three hypotheses. With all of the above in mind, it is time to reconsider the hypotheses and comment on their appropriateness.

Research Hypothesis 1

> *Object-oriented frameworks and meta-level abstraction are two complementary techniques in the design of open systems.*

As noted, no-one doubts whether object-oriented frameworks and meta-level abstractions are useful in the design of open systems as both techniques have already found their way into the design of commercial systems. So the hypothesis boils down to the question how the two techniques can complement one another.

Our work has shown that explicit representations of framework contracts are meta-objects, which implies that a potential combination of both techniques consists of a framework design with explicit objects representing framework contracts. At least this shows that the design of the base level provides clues for the initial design of the meta-level, which is an important insight in its own right.

However, this says nothing about the contracts that should be made explicit in the design. Obviously, we need some support to tell what contracts should be made explicit. Here we have proposed the four design guidelines based on the notion of a design space. Once such a design space has been devised, the four guidelines help a framework designer to identify the contracts that should be included in the meta-level. This way it is possible to incorporate the notions of domain level, system level and configuration level tailorability into the design of the framework. Moreover, we have shown that domain level tailorability can be achieved with an object-oriented framework, while system and configuration level tailorability can be achieved by meta-level abstraction. So, we conclude that both techniques are complementary in the design of open systems: *they share the common goal of tailorability, yet they differ in the level of tailorability that can be achieved*. Assuming that the three levels of tailorability correspond to an open system (see the explanation with the second research hypothesis), we claim that our first research hypothesis remains appropriate.

Research Hypothesis 2

> *Open hypermedia systems are representative for the larger set of open systems.*

The second research hypothesis was formulated because it allows us to generalise the results obtained within the context of an open hypermedia software artefact for the generic set of open systems — examples of other kinds of open systems being operating systems, database systems, inter application communication standards, tailorable software systems and programming languages.

To support the appropriateness of the hypothesis, we have argued that open hypermedia covers themes that recur in all these kinds of open systems —interoperability, extensibility and distribution. Also, we have shown that the three levels of tailorability cover those recurring themes, so we conclude that the three levels of tailorability are appropriate in the context of other kinds of open systems as well.

Then there must be a way to transpose the results of the open hypermedia software artefact to other kinds of open systems. Here we have emphasised that the definition of the three levels of tailorability is based on the notion of a design space, but is completely independent of the type and number of axes in that design space. So we conclude that, *if it is possible to devise a design space for a particular open system, then it is possible to transpose the techniques to incorporate the three levels of tailorability*. Based on other experiments (i.e., Agora and

ApplFLab) we are confident that devising a design space is feasible in other domains as well. So, assuming that it is possible to devise a design space for the application domain of the target open system, we claim that our second research hypothesis remains appropriate.

Research Hypothesis 3

Open hypermedia is a technological cornerstone for software engineering.

The third research hypothesis causes some kind of a feedback loop, in the sense that the first and second research hypotheses are about the potential benefits of applying software engineering techniques to the domain of open hypermedia systems, while the third hypothesis deals with the advantages of applying open hypermedia technology to manage the complexity of software engineering.

We have conducted various experiments in the context of what we have been calling a framework browser and we have achieved some encouraging results. First, we have shown that it is possible to use open hypermedia to connect third-party applications with the programming environment, which enables us to link source-code with design documentation in all kinds of formats. Second, we have experimented with a componentware approach for composing specific code browsers, which is advantageous for the construction of design specific views on the implementation. Third, we have examined the possibilities of an extensible link engine by incorporating specific algorithms to infer navigation relationships, which is helpful in consistency maintenance of large dynamic structures. Finally, we have explored open hypermedia as a way to publish reusable designs on the world-wide web, which is valuable in the context of reuse libraries.

The outcome of our experiments is encouraging, but there is a lot of work to be done to build a true framework browser: one that provides a seamless integration between the numerous aspects of framework development. Also, lots of ideas remain unexplored, especially concerning the support for teams. As far as we stand now, we are not able to assert the appropriateness of the third research hypothesis. Nevertheless, none of the experiments contradicted it, so we conclude that *the application of open hypermedia technology in the context of software engineering remains an interesting area for future work.*

Open Questions

Having clarified and refined some important issues in both object-oriented software engineering and open hypermedia, it is time to consider some unanswered questions and point at known limitations of our approach.

System Level Tailorability

System level tailorability is about wrapping additional behaviour around points on a design space axis and the kind of behaviour we aim at is concurrency control, logging, caching, authority control and integrity control. We propose a meta-level interface which allows us to wrap additional behaviour around the execution of framework contracts without affecting the internal implementation of the contract participants.

The typical example is maintaining a trail of all locations visited in a hypermedia session. This corresponds with wrapping logging behaviour around the execution of the navigation contract; i.e. the contract between the anchor and the resolver (see figure 23). By introducing the path object as an explicit representation of that navigation contract we are able to plug in logging behaviour without affecting the internal implementation of the anchor or resolver objects.

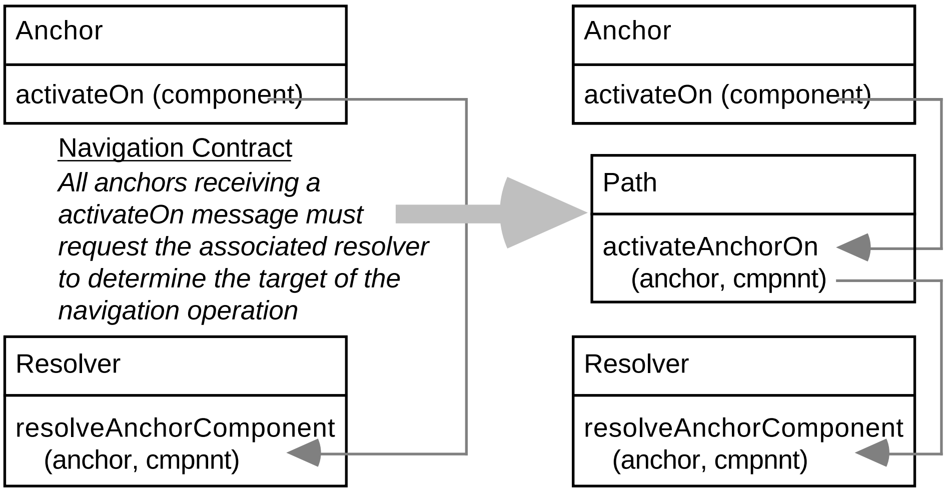

Figure 23: Logging the Navigation Contract

Actually, it is not entirely correct that the participating objects remain unaffected. Indeed, besides trapping all navigation operations one must also be able to memorise the internal state of the anchor and the component, to restore it for later use. Since both the anchor and component may represent virtual data, memorising the internal state is a complex matter. Nevertheless, the problem can be solved by applying the memento design pattern [GammaEtAl'95], which requires an extension of the interface of both the anchor and resolver objects. Such an extension may imply that one must affect the internal implementation of the participating objects, i.e. by adding extra instance variables.

Similar patterns are necessary to implement services like concurrency control, caching, authority control and integrity control; i.e. to wrap additional behaviour around an occurrence of a contract, one may need to extend the interface of the participants of that contract. However, those extensions don't have anything to do with the actual contract and are in fact situated at some other level. That is, such interface extensions must deal with issues like object identifiers, internal state, time stamps, …, which are issues that have more to do with the machine the system is running on, or the programming language the system is implemented with. Normally, such functionality is inherited from the root classes of the inheritance hierarchy.

So we must relax our assertion in the sense that one can incorporate system level tailorability by having a meta-level interface that allows to change the system behaviour, but sometimes one must make slight modifications to the external interface of base level objects. Nevertheless, the modifications to the external interface are independent compared to the extra services created and should be available for a complete session, so specified at system configuration time. In such a situation, the white-box approach is appropriate, so one can implement the extra functionality in the root classes of the inheritance hierarchy and reuse them for many kinds of services.

Singleton Meta-Objects

Another intriguing question is whether the meta-objects should remain singletons —as they are in the design of the Zypher framework— or if it is a good idea to allow multiple instances of them. There are cases supporting the viewpoint that it is better to have multiple instances.

First of all, consider a distributed setting. There, it is definitely not a good idea to work with a singleton meta-object, as such an object corresponds to a bottle-neck for the important events in the global control flow. To eliminate the bottle-neck, one may replicate the meta-object, so that there is one instance of the meta-object for each component in the distributed architecture.
So a distributed setting requires multiple instances of a meta-object. Note that all those instances will differ in internal state, but that their behaviour is essentially the same.

One may ask whether there exist situations that require multiple instances of meta-objects, where those instances will differ in both state and behaviour. Considering the schema in figure 23, suppose one wants to log all activations of anchors created after a certain date and leave the activations of all other anchors out of the log. One solution would be to include some filter in the log algorithm implemented in the path object. Another solution would be to have two path objects with different behaviour; one that writes its activations to the log and another one that does not. The filter is then embedded in a separate dispatcher object that decides what objects should be managed by the first path object, and what should be managed by the second one.
Although the first solution is easier to implement, the second solution is more flexible as it corresponds to a black-box approach. For example, as the filter algorithm is independent of the path object, the second solution makes it possible to reconfigure the logging algorithm at run-time.

Having multiple instances of meta-objects that differ in both state and behaviour is more flexible, yet this flexibility comes at a cost. As long as there is only one meta-object —or multiple replicas of one and the same object—, one can maintain a reference to this meta-object in some global variable. Once there exist multiple instances that differ in both state and behaviour, one must include some reference mechanism from the base level object to the meta-object that is supposed to handle the meta-level behaviour. Such reference mechanism may be quite complicated, as it may depend on a particular combination of base level objects. For example, the path meta-object may differ depending on the particularanchor-component-resolver triple that participates in the navigation contract.

Additional Levels of Tailorability

We have proposed three levels of tailorability —domain level, system level and configuration level— and have argued that they correspond to respectively a framework base level, a meta-level and a meta-meta-level. The levels of tailorability covered the open hypermedia requirement list, which implies that we need at least those three levels of tailorability. However, one can ask if there are other useful levels of tailorability and whether they would correspond with higher meta-levels as well.

We don't have an answer to this question, but we do recognise two viable candidates for additional levels. A first one is about adding or removing axes to a design space. As we claim that the levels of tailorability are independent of the type and number of axes, it should be possible to include an interface to change the axes system. And somehow this would correspond with a higher meta-level, because it must necessarily change the meta-meta-object representing the configuration of the design space.

A second one is about the configuration of the meta-objects. In the current design, there is only one instance of each meta-object in the system, but —as argued above— some situations may require multiple instances of the same meta-object. And then the decision about what instance of a meta-object should handle what base level object is crucially important. The influence this decision an additional level of tailorability is required. This would correspond with a higher meta-level, because it is about the configuration of the meta-objects.

Additional Hypermedia Design Space Axes

We proposed our perspective on hypermedia as a three dimensional design space, the three dimensions being storage, presentation and navigation. We argued that the name and number of the dimensions does not affect the remainder of our argumentation, this way avoiding the discussion if those three dimensions are adequate for describing the space of hypermedia systems. In fact, it remains an open question whether the three dimensions are necessary and sufficient.

We are quite confident that a hypermedia design space should at least include those three dimensions. First, because all the dimensions are somehow included in the hypertext & hypermedia definition (p.08) we adopted; second, because the experiments with the framework browser have shown that the three dimensions are at least appropriate. Nevertheless, we don't have any arguments for supporting whether the three dimensions are sufficient.

An interesting candidate for an extra dimension would be the authoring of navigation structures. Authoring is an aspect that is not made explicit in our design, because we have focused on navigation structures that can be derived from the contents. So, in the Zypher prototype, authoring is part of the navigation dimension, which implies that each point on the navigation axis (i.e. each resolver) defines its own authoring facilities. An experiment that may provide some answers would be to capture a hypermedia system that emphasises on the authoring of hypermedia structures —such as the spatial hypermedia systems Aquanet and VIKI— in the current design of the Zypher framework.

Black-box versus White-box

The argumentation for incorporating the three levels of tailorability into the design of a framework was about favouring a black-box approach over a white-box approach, or in other words object composition over class inheritance. One may ask what would happen if for some reason the white-box approach was favoured over the black-box approach.

Consider the navigation axis of our hypermedia design space (see also figure 23). To incorporate domain level tailorability, we need to be able to vary the resolution of the anchor independently of its activation, which implies that the second design guideline would be rephrased as "each design space axis should correspond to a white-box template method". In the white-box reuse approach, this would mean that we have a single anchor object that understands both the `activateOn` and `resolveAnchorComponent` messages, but that both of them may be inherited from different classes.

To incorporate system level tailorability on the navigation axis, we must be able to wrap additional behaviour around the execution of the `activateOn` and `resolveAnchorComponent` methods. In a white-box approach this means that we must inherit the wrapping behaviour from a common superclass of the anchor classes or the resolver classes involved. To ensure this inheritance behaviour in the framework, the framework must include a contract stating that anchor classes and resolver classes must delegate the navigation messages to their superclass; i.e. they must perform a super send. So the third design guideline, stating that "each design space axis should correspond to a framework contract" remains valid, but the way this design guideline is interpreted in the design of the framework is different.

To incorporate configuration level tailorability in a white-box approach, we must be able to extend the inheritance hierarchy of the framework; preferably using multiple inheritance. For instance, if we want to have a relationship between a particular point on the navigation axis and a particular point on the storage axis, we will create an anchor class that inherits from a class implementing the behaviour of that storage device and a class that implements that particular link resolution algorithm. However, in that case the inheritance hierarchy becomes an essential aspect of the framework design, which implies that it should be captured in a framework contract. So, the fourth design guideline, stating that "the configuration of the design space axes should correspond to a framework contract" remains valid, but again the way this design guideline is interpreted in the design of the framework is different.

The main argument for favouring the black-box approach over white-box approach is that this makes it easy to reconfigure the framework at run-time: one can change the execution of black-box template methods by changing the object composition. In a white-box approach, run-time reconfiguration of the framework corresponds with run-time modifications to the inheritance hierarchy. There exist object-oriented languages (Smalltalk and CLOS to mention the most popular ones) that allow to modify the class hierarchy at run-time, however this functionality always belongs to the meta-object protocol of the programming language. Indeed, the class hierarchy is about the software system in a causally connected way.

Note that the white-box approach —even with the possibility of run-time framework reconfiguration— remains less flexible than the black-box approach, because the black-box approach allows to reconfigure the framework at object level, while the white-box approach is restricted to reconfigurations at the class level. Reconfiguration at object level in a white-box approach would imply the possibility of changing the class of an existing object at run-time which is a very difficult operation at it requires changes to the objects internal state representation (i.e. the set of instance variables) and behaviour (i.e. the method table). Anyway, such an operation would also belong to the meta-object protocol of the programming language, as the object-class relationship is about the software system in a causally connected way.

The above argumentation leads us to the conclusion that favouring a white-box approach over a black-box approach would not change that much. To incorporate the three levels of tailorability, the design guidelines we propose remain essentially the same; only the way they should be interpreted in the design of the framework changes. And if one wants to reconfigure the framework at run-time, we need a meta-object protocol that is part of the programming language used to implement the framework.

Afterthought

As an final remark, we emphasise on the cross-fertilisation embodied in our work. On the one hand we build on the accumulated experience of the software engineering community. More precisely we deal with techniques studied in very technological areas like programming languages, distributed architectures and operating systems. As a consequence, the techniques are well specified and their effects quite accurate, yet they are seldom applied in other applications domains. On the other hand, our work builds on the insights of hypermedia, a domain with years of fruitful research on how to enhance the way people deal with information. The problems encountered there are rather fuzzy and the solutions quite imprecise, yet hypermedia research has changed the working habits of almost all people working with computers.

Our contribution is original in the way it captures the best of both worlds: the precision of software engineering with the applicability of hypermedia. And this idea reveals the heart of what we entitled "the link from object-oriented software engineering to open hypermedia".

CHAPTER 10

Appendices

References

[Abelson,Sussman'84]
Abelson, H. / Sussman, G. J. "Structure and Interpretation of Computer Programs"; MIT Press, 1984.

[Akscyn'91]
Akscyn, R. "Design Tradeoffs for Advanced Hypertext Technology"; Tutorial Notes at [HT'91].

[Akscyn,McCracken,Yoder'87]
Akscyn, R. / McCracken, D. / Yoder, E. "KMS: A Distributed Hypermedia System for Managing Knowledge in Organisations"; in [HT'87]. Republished in Communications of the ACM - Vol. 31(7) - July '88.

[AksitEtAl'93]
Aksit, M. / Wakita, K. / Bosh, J. / Bergmans, L. / Yonezawa, A. "Abstracting Object Interactions Using Composition Filters"; ECOOP '93, Workshop on Object-Based Distributed Programming - Lecture Notes in Computer Science, nr. 791 – Berlin Heidelberg, July 1993. Available on the world-wide web via"ftp://ftp.cs.utwente.nl/pub/doc/TRESE/AbstrObjIntUsingCF.ps.Z".

[Anderson,Taylor,Whitehead'94]
Anderson, K. M. / Taylor, R. N. / Whitehead, E. J. Jr. "Chimera: Hypertext for Heterogeneous Software Environments"; in [ECHT'94].

[Andrews,Kappe,Maurer'95]
Andrews, K. / Kappe, F. / Maurer, H. "Serving Information to the Web with Hyper-G"; Proceedings of the Third International world-wide web Conference - WWW'95; Computer Networks and ISDN Systems - Vol. 27(6) - April 1995.

[Apple'89]
Apple Computer, Inc. "Macintosh HyperCard User's Guide"; Apple Computer, Inc., 1989.

[Ashman'94]
Ashman, H. "Opinion: What is Hypermedia"; ACM SIGLINK Newsletter - Vol. 3(2) - September '94.

[Aßfalg'94]
Aßfalg, R (Ed) "Open Hypertext Systems" at the Universität Konstanz, 26/27 May 1994. Send e-mail to Rainer Kuhlen (kuhlen@inf-wiss.uni-konstanz.de) or Marc Rittberger (ritt@inf-wiss.uni-konstanz.de).

[Beck,Johnson'94]
Beck, K. / Johnson, R. "Patterns Generate Architectures"; ECOOP'94 Proceedings, Lecture Notes in Computer Science nr. 821, Springer-Verlag, 1994. Available on the world-wide web via http://st-www.cs.uiuc.edu /users /patterns /patterns.html.

[Berners-LeeEtAl'94]
Berners-Lee, T. / Cailliau, R. / Luotonen, A. /

Nielsen, H. F. / Secret, A. "The world-wide web"; Communications of the ACM - Vol. 37(8) - August '94.

[Birrer,Eggenschwiler'93]
Birrer, A. / Eggenschwiler, T. "Frameworks in the Financial Engineering Domain: An Experience Report"; ECOOP'93 Proceedings, Lecture Notes in Computer Science nr. 707, Springer-Verlag, 1993.

[Blaha,Premerlandi,Rumbaugh'88]
Blaha, M. R. / Premerlani, W. J. / Rumbaugh, J. E. "Relational Database Design Using an Object-Oriented Methodology"; Communications of the ACM - Vol. 31(4) - April '88.

[Brooks'87]
Brooks, F. P. Jr. "No Silver Bullet: Essence and Accidents in Software Engineering"; IEEE Computer - April '87.

[Bush'45]
Bush, V. "As we May Think"; Atlantic Monthly - 176 - July, 1945. Available on the world-wide web via http://www.isg.sfu.ca/~duchier/misc/vbush/.

[Campbell,Goodman'87]
Campbell, B. / Goodman, J. M. "HAM: A General - Purpose Hypertext Abstract Machine"; in [HT'87]. Republished in Communications of the ACM - Vol. 31(7) - July '88.

[CampbellEtAl'93]
Campbell, R. H. / Islam, N. / Raila, D. / Madeany, P. "Designing and Implementing Choices: An Object-oriented System in C++"; Communications of the ACM - Vol. 36(9) - September'93.

[Catlin,Bush,Yankelovich'89]
Catlin, T. / Bush, P. / Yankelovich, N. "InterNote: Extending a Hypermedia Framework to Support Annotative Collaboration"; in [HT'89].

[CodenieEtAl'94]
Codenie, W. / De Hondt, K. / D'Hondt, T. / Steyaert, P. "Agora: Message Passing as a Foundation for Exploring OO Language Concepts"; ACM SIGPLAN Notices, Vol. 29(12), December '94.

[Conklin'87]
Conklin, J. "Hypertext: An Introduction and Survey"; IEEE Computer - Vol. 20 (9) - September 1987.

[Conklin,Begeman'88]
Conklin, J. / Begeman, M. L. "gIBIS: A Hypertext Tool for Exploratory Policy Discussion"; CSCW'88 Proceedings, ACM Press, 1988.

[Cotter,Potel'95]
Cotter, S. / Potel, M. "Inside Talingent Technology"; Addison-Wesley, 1995.

[Davis'95]
Davis, H. "To Embed or Not to Embed …"; Communications of the ACM - Vol. 38(6) - August '95. Available on the world-wide web via http://www.acm.org/siglink/.

[Davis,Knight,Hall'94]
Davis, H. C. / Knight, S. / Hall, W. "Light Hypermedia Link Services: A Study of Third Party Application Integration"; in [ECHT'94].

[Davis,Lewis,Rizk'96]
Davis, H. / Lewis, A. / Rizk, A. "OHP: A Draft Proposal for a Standard Open Hypermedia Protocol"; in [Wiil,Demeyer'96]

[DavisEtAl'92]
Davis, H. / Hall, W. / Heath, I. / Hill, G. / Wilkins, R. "Towards An Integrated Information Environment with Open Hypermedia Systems"; in [ECHT'92].

[DeBra,Houben,Kornatzky'92]
De Bra, P. / Houben, G. J. / Kornatzky, Y. "An Extensible Data Model for Hyperdocuments"; in [ECHT'92].

[DuvalEtAl95]
Duval E. / Olivie H. / O'Hanlon P. / David G. Jameson "HOME: an Environment for Hypermedia Objects"; Journal of Universal Computer Science - Vol 1(5) - May 1995. Available on the world-wide web via http://www.iicm.tu-graz.ac.at/Cjucs_root. Also "HOME: Hypermedia Object Management Environment"; Demonstration at ECHT'94.

[ECHT'90]
European Conference on Hypertext'90, November 27-30, Paris, France. Proceedings: Rizk, A. / Streitz, N. / André, J. "Hypertext: Concepts, Systems and Applications - Proceedings of the European Conference on Hypertext", Cambridge University Press, 1990.

[ECHT'92]
European Conference on Hypertext'92, November 30 - December 4, Milano, Italy. Proceedings: ACM Press, 1992.

[ECHT'94]
European Conference on Hypertext'94, September 18-23, Edinburgh, UK. Proceedings: ACM Press, 1994.

[Engelbart'95]
Engelbart, D. C. "Toward Augmenting the Human Intellect and Boosting our Collective IQ"; Communications of the ACM - Vol. 38(8) - August '95. Available on the world-wide web via http://www.acm.org/siglink/.

[Forman,Danforth,Madduri'95]
Forman, I. R. / Danforth, S. / Madduri, H. "Composition of Before/After Metaclasses in SOM"; AIXpert, May 1995.

[FountainEtAl'90] Fountain, A. M. / Hall, W. / Heath, I. / Davis, H. C. "MICROCOSM: An

Open Model for Hypermedia With Dynamic Linking"; in [ECHT'90].

[Furuta,Stotts'89]
Furuta, R. / Stotts, P. D. "Programmable Browsing Semantics in Trellis"; in [HT'89].

[GammaEtAl'95]
Gamma, E. / Helm R. / Johnson R. / Vlissides, J. "Design Patterns"; Addisson-Wesley, 1995. The same people have written the paper "Design Patterns: Abstraction and Reuse of Object-Oriented Design"; ECOOP'93 Proceedings, Lecture Notes in Computer Science nr. 707, Springer-Verlag, 1993.

[Goldberg'84]
Goldberg, A. "Smalltalk-80, the Interactive Programming Environment"; Addison-Wesley, 1984.

[Goldberg,Rubin'95]
Goldberg, A. / Rubin, K. S. "Succeeding with Objects. Decision Frameworks for Project Management"; Addison-Wesley, 1995.

[Grønbaek'94]
Grønbaek, K. "Composites in a Dexter-Based Hypermedia Framework"; in [ECHT'94].

[Grønbaek,Malhotra'94]
Grønbaek, K. / Malhotra, J. "Building Tailorable Hypermedia Systems: the embedded-interpreter approach"; OOPSLA '94 Proceedings - ACM Press, 1994.

[Grønbaek,Trigg'94]
Grønbaek, K. / Trigg, R. H. "Design issues for a Dexter-Based Hypermedia System"; Communications of the ACM - Vol. 37(2) - February '94. Also in [ECHT'92].

[Grønbaek,Trigg'96]
Grønbaek, K. / Trigg, R. H. "Towards a Dexter-Based model for Open Hypermedia: Unifying Embedded References and Link Objects"; in [HT'96].

[GrønbaekEtAl'94]
Grønbaek, K. / Hem, J. A. / Madsen, O. L. / Sloth, L. "Systems: A Dexter-based Architecture"; Communications of the ACM - Vol. 37(2) - February '94. Also in [HT'93] under the title "Designing Dexter-based Cooperative Hypermedia Systems".

[HaanEtAl'92]
Haan, B. J. / Kahn, P. / Riley, V. A. / Coombe, J. H. / Meyrowitz, N. K. "IRIS Hypermedia services"; Communications of the ACM - Vol. 35(1) - January '92.

[Halasz'87]
Halasz, F. G. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems"; in [HT'87]. Republished in Communications of the ACM - Vol. 31(7) - July '88.

[Halasz'91]
Halasz, F. G. "Seven Issues: Revisited"; Keynote Address at [HT'91]. The slides and a transcript of the speech are available on the world-wide web via http://www.parc.xerox.com/spl/projects/halasz-keynote/transcript.html.

[Halasz,Schwartz'90]
Halasz, F. / Schwartz, M. "The Dexter Hypertext Reference Model"; Proceedings of the 1990 NIST Hypertext Standardisation Workshop (January 16-18, Gaithersburg, MD). Republished in Communications of the ACM - Vol. 37(2) - February '94.

[Hall,Hill,Davis'93]
Hall, W. / Hill, G. / Davis, H. "The Microcosm Link Service"; in [HT'93].

[Helm,Holland,Gangopadhyay'90]
Helm, R. / Holland, I. M. / Gangopadhyay, D. "Contracts: Specifying Behavioral Compositions in Object-Oriented Systems"; ECOOP/OOPSLA'90 Proceedings, ACM Press, 1990.

[Hill,Hall'94]
Hill, G. / Hall, W. "Extending the Microcosm Model to a Distributed Environment"; in [ECHT'94].

[HT'87]
Hypertext'87, November 13-15, Chapel Hill, North Carolina. Proceedings: ACM Press, 1987. Some of the papers were republished in Communications of the ACM - Vol. 31(7) - July '88.

[HT'89]
Hypertext'89, November 5-8, Pittsburgh, Pennsylvania. Proceedings, ACM Press, 1989.

[HT'91]
Hypertext'91, December 15-18, San Antonio, Texas. Proceedings: ACM Press, 1991.

[HT'93]
Hypertext'93, November 14-18, Seattle, Washington. Proceedings: ACM Press, 1993.

[HT'96]
Hypertext'96, March 16-20, Washington DC. Proceedings: ACM Press, 1996.

[JacobsonEtAl'92] Jacobson, I. / Christerson, M. / Jonsson, P. / Overgaard, G. "Object-Oriented Software Engineering - A Use Case Driven Approach; Addison-Wesley, 1992.

[Java'95]
Java "The HotJava Browser: A White Paper"; available on the world-wide web via http://java.sun.com/.

[Johnson'92]
Johnson, R. "Documenting Frameworks Using Patterns"; OOPSLA'92 Proceedings, ACM Press, 1992. Available on the world-wide web via http://st-www.cs.uiuc.edu /users /patterns /patterns.html.

[Johnson,Foote'88]
Johnson, R. E. / Foote, B. "Designing Reuseable Classes"; Journal of Object-Oriented Programming - Vol 1(2) - February 1988.

[Johnson,Russo'91]
Johnson, R. E. / Russo, V. F. "Reusing Object-Oriented Designs"; University of Illinois Tech Report UIUICDCS 91-1696.

[Kacmar,Leggett'91] Kacmar, C. J. / Legget, J. J. "PROXHY: A Process-Oriented Extensible Hypertext Architecture"; ACM Transactions on Information Systems, Vol. 9 (4), October '91.

[Kiczales'94a]
Kiczales, G. "Why are Black Boxes so Hard to Reuse ? Toward a New Model of Abstraction in the Engineering of Software"; Keynote address at OOPSLA'94. Available on the world-wide web via http://www.parc.xerox.com/spl/projects/oi/towards-talk/transcript.htm.

[Kiczales'94b]
Kiczales, G. (Ed.) "Foil For The Workshop On Open Implementation"; Xerox Parc. Available on the world-wide web via http://www.parc.xerox.com/spl/projects/oi/workshop-94/foil/main.html.

[Kiczales,Rivières,Bobrow'91]
Kiczales, G. / des Rivières, J. / Bobrow, D. G. "The Art of the Metaobject Protocol"; The MIT Press, 1991.

[Knopik,Bapat'94]
Knopik, T. / Bapat, A. "The Role of Node and Link Types in Open Hypermedia Systems" in [Wiil,Østerbye'94].

[Lai,Manber'91]
Lai, P. / Manber, U. "Flying Through Hypertext"; in [HT'91].

[Leggett,Schnase'94]
Leggett, J. J. / Schnase, J. L. "Viewing Dexter with Open Eyes"; Communications of the ACM - Vol. 37(2) - February '94.

[LewisEtAl.'96]
Lewis, P. H. / Davis, H. C. / Griffiths, S. R. / Hall, W. / Wilkins, R. J. "Media-Based Navigation with Generic Links"; in [HT'96].

[Lucas,Steyaert,Mens'95] Lucas, C. / Mens, K. / Steyaert, P. "Typing Dynamic Inheritance, a Trade-Off between Substitutability and Extensibility. Technical Report vub-prog-tr-95-03.

[Maes'87]
Maes, P. "Concepts and Experiments in Computational Reflection"; OOPSLA'87 Proceedings, ACM Press, 1987.

[Malcolm,Poltrock,Shuler'91]
Malcolm, K. C. / Poltrock, S. E. / Schuler, D. "Industrial Strength Hypermedia: Requirements for a Large Engineering Entreprise"; in [HT'91].

[Marshall,Shipman'93]
Marshall, C. C. / Shipman, F. M. "Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext"; in [HT'93].

[Marshall,Shipman'95]
Marshall, C. C. / Shipman, F. M. "Spatial Hypertext: Designing for Change"; Communications of the ACM - Vol. 38(8) - August '95. Available on the world-wide web via http://www.acm.org/siglink/.

[Marshall,Shipman,Coombs'94]
Marshall, C. C. / Shipman, F. M. / Coombs, J. H. "VIKI: Spatial Hypertext Supporting Emergent Structure"; in [ECHT'94].

[MarshallEtAl'91]
Marshall, C.C. / Halasz, F. G. / Rogers, R. A. / Janssen, W. C. "Aquanet: A Hypertext Tool to Hold Your Knowledge in Place"; in [HT'91].

[McAffer'95]
McAffer, J. "Meta level Programming with CodA"; ECOOP'95 Proceedings, Lecture Notes in Computer Science nr. 952, Springer-Verlag, 1995.

[Meyrowitz'86]
Meyrowitz, N. "Intermedia: The Architecture and Construction of an Object-Oriented Hypermedia System and Applications Framework"; OOPSLA'86 Proceedings, ACM Press, 1986.

[Meyrowitz'89]
Meyrowitz, N. "The missing link: Why we're all doing hypertext wrong"; in Barrett, E. (ed) "The Society of Text", The MIT Press, Cambridge, 1989.

[Mylonas,Heath'90]
Mylonas, E. / Heath, S. "Hypertext from the Data Point of View: Paths and Links in the Perseus Project"; in [ECHT'90].

[NationalAcademy'94] National Academy of Sciences "Academic Careers for Experimental Computer Scientists and Engineers"; Communications of the ACM, Vol. 37(2), April '94.

[Nelson'87]
Nelson, T. H. "All for One and One for All"; in [HT'87].

[Newcomb,Kipp,Newcomb'91]
Newcomb, S. R. / Kipp, N. A. / Newcomb, V. T. "Hytime: Hypermedia / Time-based Document Structuring Language"; Communications of the ACM, Vol. 34(11), November '91.

[Nielsen'90]
Nielsen, J. "Hypertext & Hypermedia"; Academic Press, 1990.

[NürnbergEtAl'96]
Nürnberg, P. J. / Leggett, J. J. / Schneider, E. R. / Schnase, J. L. "Hypermedia Operating Systems: A New Paradigm for Computing"; in [HT'96].

[Østerbye,Wiil'96]
Østerbye, K. / Wiil, U. K. "The Flag Taxonomy of Open Hypermedia Systems"; Hypertext '96 Proceedings, ACM Press, 1996. See also [Wiil,Demeyer'96].

[Paepcke'90]
Paepcke, A. "PCLOS: stress testing CLOS"; ECOOP/OOPSLA'90 Proceedings, ACM Press 1990.

[Parunak'91]
Parunak, H.V.D. "Don't Link Me In: Set Based Hypermedia for Taxonomic Reasoning"; in [HT'91].

[Pearl'89]
Pearl, A. "Sun's Link Service: A Protocol for Open Linking"; in [HT'89].

[Poltock'96]
Poltrock, S. E. "Keynote address at the 2nd Workshop on Open Hypermedia Systems — Hypertext '96, Washinton D.C." See [Wiil,Demeyer'96].

[Pree'94]
Pree, W. "Design Patterns for Object-Oriented Software Development"; Addison-Wesley 1994.

[Rao'91]
Rao, R. "Implementational Reflection in Silica"; ECOOP'91 Proceedings, Lecture Notes in Computer Science, P. America (Ed.), Springer-Verlag, 1991.

[Remde,Gomez,Landauer'87]
Remde, J. R. / Gomez, L. M. / Landauer, T. K. "SuperBook: An Automatic Tool for Information Exploration — Hypertext ?"; in [HT'87].

[Rizk,Sauter'92]
Rizk, A. / Sauter, L. "Multicard: An Open Hypermedia System"; in [ECHT'92].

[RumbaugEtAl'91]
Rumbaugh, J. Blaha, M. / Premerlani, W. / Eddy, F. / Lorensen, W. "Object-Oriented Modeling and Design"; Prentice Hall, 1991.

[Sawhney,Balcom,Smith'96]
Sawhney, N. / Balcom, D. / Smith, I. "HyperCafe: Narrative and Aesthetic Properties of HyperVideo"; in [HT'96].

[Schackelford,Smith,Smith'93]
Schackelford, D. E. / Smith, J.B. / Smith, F.D, "The Architecture and Implementation of a Distributed Hypermedia Storage System"; in [HT'93].

[Schmucker'86]
Schmucker, K. "Object-Oriented Programming for the Macintosh"; Hayden Book Company, 1986.

[Schütt,Streitz'90]
Schütt, H. / Streitz, N. "HyperBase: A Hypermedia Engine Based on a Relational Data-Base Management System"; in [ECHT'90].

[Smith'96]
Smith, J.B. "Invited statement at the 2nd Workshop on Open Hypermedia Systems — Hypertext '96, Washinton D.C." See [Wiil,Demeyer'96].

[Smith,Smith'91]
Smith, J. B. / Smith, F. D. "ABC: A Hypermedia System for Artifact-Based Collaboration"; in [HT'91].

[Smith,Zdonik'87]
Smith, K.E. / Zdonik, S. B. "Intermedia: A Case Study of the Differences Between Relational and Object-Oriented Database Systems"; OOPSLA'87 Proceedings, ACM Press, 1987.

[Steyaert'94]
Steyaert, P. "Open Design of Object-Oriented Languages"; Phd. dissertation, Vrije Universiteit Brussel, 1994. Available on the world-wide web via http://progwww.vub.ac.be/prog/papers/paperquery.

[Steyaert,DeMeuter'95]
Steyaert, P. / De Meuter, W. "A Marriage of Class- and Object-Based Inheritance Without Unwanted Children"; ECOOP'95 Proceedings, Lecture Notes in Computer Science nr. 952, Springer-Verlag, 1995. Available on the world-wide web via http://progwww.vub.ac.be/prog/papers/paperquery.

[SteyaertEtAl'93]
Steyaert, P. / Codenie, W. / D'Hondt, T. / Lucas, C. / Van Limberghen, M. "Nested Mixin-Methods in Agora"; ECOOP'93 Proceedings, Lecture Notes in Computer Science nr. 707, Springer-Verlag, 1993. Available on the world-wide web via http://progwww.vub.ac.be/prog/papers/paperquery.

[SteyaertEtAl'94]
Steyaert, P. / De Hondt, K. / Demeyer, S. / De Molder, M. "A Layered Approach To Dedicated Application Builders Based On Application Frameworks"; In Patel D. / Sun, Y. / Patel, S. (Ed) "Proceedings of the 1994 International Conference on Object-Oriented Information Systems"; Springer-Verlag, 1995. Available on the world-wide web via http://progwww.vub.ac.be/prog/papers/paperquery.

[SteyaertEtAl'96]
Steyaert, P. / De Hondt, K. / Demeyer, S. / Boyen, N. "Reflective Application Builders"; In Chris Zimmermann (Ed) "Advances in Object-Oriented Metalevel Architectures and Reflection"; CRC Press Inc., Boca Raton, Florida, 1996. Available on the world-wide web via http://progwww.vub.ac.be/prog/papers/paperquery.

[Trigg'96]
Trigg, R. "Hypermedia as Integration: Recollections, reflections and exhortations"; Keynote Address at [HT'96].

[Udell'94]
Udell, J. "Componentware"; BYTE- May '94.

[VanLimberghen'96]
Van Limberghen, M. "Building Frameworks Through Specialisable Nested Objects"; to appear in Tools-USA'96 Proceedings, Prentice Hall, 1996. Available on the world-wide web via http://progwww.vub.ac.be/prog/papers/paperquery.

[Vanzyl'94]
Vanzyl, A. J. "Open Hypermedia Systems: Comparisons and Suggestions for Implementation Strategies" in [Wiil,Østerbye'94].

[WeinandEtAl'88]
Weinand, A. / Gamma, E. / Marty, R. "ET++, An Object-Oriented Application Framework in C++"; OOPSLA'88 Proceedings, ACM Press, 1988.

[Wiil,Demeyer'96]
Wiil, U. K. / Demeyer, S. (Ed) "Proceedings of the 2nd Workshop on Open Hypermedia Systems - Hypertext'96"; UCI-ICS Technical Report 96-10. Department of Information and Computer Science, University of California, Irvine, CA 92717-3425. See also http://www.iesd.auc.dk/~kock/OHS-HT96/.

[Wiil,Leggett'92]
Wiil, U. K. / Leggett, J. J. "Hyperform: Using Extensibility to Develop Dynamic, Open and Distributed Hypertext Systems"; in [ECHT'92].

[Wiil,Leggett'93]
Wiil, U. K. / Leggett, J. J. "Concurrency Control in Collaborative Hypertext Systems"; in [HT'93].

[Wiil,Leggett'96]
Wiil, U. K. / Leggett, J. J. "The HyperDisco Approach to Open Hypermedia Systems"; in [HT'96].

[Wiil,Østerbye'94]
Wiil, U. K. / Østerbye, K. (Ed) "Proceedings of the ECHT'94 Workshop on Open Hypermedia Systems"; Technical report R-94-2038 / Institute for Electronic Systems Department of Mathematics and Computer Science - Fredrik Bajers Vej 7 - DK 9220 Aalborg - Denmark. Available on the world-wide web via ftp://ftp.iesd.auc.dk/pub/packages/hypertext/ECHT94-workshop/.

[Wirfs-Brock,Johnson'90]
Wirfs-Brock, R. J. / Johnson, R. E. "Surveying current Research in Object-Oriented Design"; Communications of the ACM - Vol. 33(9) - September'90.

[Wirth'76]
Wirth, N. "Algorithms+Data Structures=Programs"; Prentice Hall,76.

[Wuyts'96]
Wuyts, R. "Class-Management using Logical Queries, Application of a Reflective User Interface Builder"; GRONICS'96 Proceedings of the third Groningen International Information Technology Conference for Students; University of Groningen, 1996. Available on the world-wide web via http://progwww.vub.ac.be/prog/papers/paperquery.

[Yankelovich,Meyrowitz,VanDam'85]
Yankelovich, N. / Meyrowitz, N. / van Dam, A. "Reading and Writing the Electronic Book"; IEEE Computer - October '85.

[Yokote'92]
Yokote, Y. "The Apertos Reflective Operating System: The Concept and Its Implementation"; OOPSLA'92 Proceedings, ACM Press, 1992.

[Yokote,Teraoka,Tokoro'89]
Yokote, Y. / Teraoka, F. / Tokoro, M. "A reflective architecture for an object-oriented distributed operating system"; ECOOP'89 Proceedings, Cambridge University Press, 1989.

[Zellweger'89]
Zellweger, P. T. "Scripted Documents: A Hypermedia Path Mechanism"; in [HT'89].

Index

List of Figures

Detailed Table of Contents

Aposition / Bijstelling

Group decision processes on the world-wide web must be supported by structured communication to improve information exchange and unstructured communication to cope with meta-discourse.

Explanation

With the appearance of the world-wide web [Berners-LeeEtAl'94], people are confronted with a tremendous increase in communication facilities. However, extra communication facilities do not necesarily improve information exchange. This phenomenon can be observed in discussions on bulletin board systems, newsgroups and mail lists: such discussions quickly lose focus and rarely come to a conclusion. If they come to a conclusion, it is because the group managed to *structure its communication process*, for example by arranging a vote.

Group decision support system (GDSS) research is especially concerned with the question how structured communication processes improve decision making. GDSS like GroupSystems [NunamakerEtAl'91], COPE [Eden,Ackermann'92] and DSide [Kenis'95], [KenisEtAl'95] show that by increasing the number of alternatives considered, enhancing the depth of analysis each alternative is considered with and enlarging the participation of all group members the quality of the decision process improves. Nevertheless, studies revealed that individual group members often feel constrained by the structure, among others for not supporting meta-discourse[11] [Conklin,Begeman'88]. Hence the requirement for some kind of *unstructured communication* between participants.

In a synchronous-proximate setting (i.e. all group members gather at the same time, in the same place), unstructured communication is usually accomplished by some form of verbal contact (mostly through the facilitator). However, experiments that abandon the meeting room can not rely on verbal contact and need other unstructured communication channels. During an experiment with the DSide version accessible via the world-wide web [KenisEtAl'96] we found out that participants used e-mail as a natural communication channel for meta-discourse, allthoug e-mail was not an explicit part of the DSide system.

The use of unstructured communication for meta-discource can be observed in other experiments as well. GroupSystems has been tested in a synchronous-remote setting [NunamakerEtAl'91] and their unstructured communication was available under the form of video conferencing. COPE has been tested in an asynchronous-proximate setting [Trahand'93], but members discussed the decision graph afterwards.

[Berners-LeeEtAl'94] Berners-Lee / Cailliau, R. / Luotonen, A. / Nielsen, H. F. / Secret, A. "The world-wide web"; Communications of the ACM - Vol. 37(8) - August '94.

11 Meta-discourse, as opposed to communication about the topic, is communication about the communication process.

[Eden,Ackermann'92] Eden, C. / Ackerman, F. "Strategy Development and Implementation - The Role of a Group Decision Support System"; in "Computer Augmented Teamwork: A Guided Tour" (ed. Bostrom, R. P.; Watson, R. T.; Kinney, S. T.); Van Nostrand Reingold, 1992.

[Kenis'95] Kenis, D. "Improving Group Decisions: Designing and Testing Techniques for Group Decision Support Systems Applying Delphi Principles"; PhD. at the University of Utrecht, 1995.

[KenisEtAl'95] Kenis, D. / Demeyer, S. / Maréshall, T. / Boyen, N. "Concluding Session using the DSide GDSS System"; EURO GDSS Workshop, Tilburg University, Netherlands; 20/21st April 1995.

[KenisEtAl'96] Kenis, D. / Lybaert, W. / Maréshall, T. "D-Side: a Worl-Wide Web Based GDSS"; EURO GDSS Workshop, Grenoble, France; 27/28 March 1995.

[Trahand'93] Trahand, J. "Structured Electronic Conversations"; Presentation at the European GDSS and Negotiation Workshop; Strathclyde Graduate Business School; 4/5th November 1993.

[Conklin,Begeman'88] Conklin, J. / Begeman, L. "gIBIS: A Hypertext Tool for Exploratory Policy Discussion"; CSCW'88 Proceedings, ACM Press, 1988.

[NunamakerEtAl'91] Nunamaker, J.F. / Dennis. A. R. / Valachi, J. S. / Vogel, D. R. / George, J. F. "Electronic Meeting Systems to support Group Work"; Communications of the ACM, Vol. 34(7), July '91.