L. Calvi & A. Cristea, Towards Generic Adaptive Systems: Analysis of a Case Study, AH 2002, Adaptive Hypermedia & Adaptive Web-Based Systems, LNCS 2347, Springer, 79-89

Towards Generic Adaptive Systems: Analysis of a Case Study

Licia Calvi and Alexandra Cristea

Information System Department, Technical University Eindhoven, The Netherlands
{a.i.cristea, l.calvi}@tue.nl
http://wwwis.win.tue.nl/~{calvi, alex}

Abstract. This paper analyses one of the most well-known general purpose adaptive hypermedia systems, AHA!, and, based on this analysis, make some general observations about adaptive hypermedia systems and some improvement suggestions for the AHA! system. We suggest here a concept-based approach to the structuring of adaptive hypermedia systems, as well as an extension of the well-known rule-based overlay method for user-adaptation. This approach is another step towards flexible generic-purpose adaptive hypermedia.

1 Introduction

Adaptation in hypermedia has been traditionally divided [3] into *adaptive navigation support* (link level adaptation) and *adaptive presentation* (content level adaptation). However, this division causes problems at a conceptual level, which can lead, from an authoring point of view, to difficulties in the definition of concept relationships.

Defining which links to show to users and which *concept granularity* to associate to those links presupposes a correlation between concepts and links that is not directly matched onto the simple link visibility function adopted by most adaptive systems. Learning is indeed always situated, it never occurs in the vacuum. In a situated learning scenario, concepts interact with one another to the extent that one concept may assume a meaning on the basis of the context it belongs to, i.e., of the concepts that surround it. The influence of contextualization on the learning process is more evident if we think of the adaptive system as a tool to enhance knowledge acquisition.,

In this view, concepts and links need to be intertwined in order to allow authors to distinguish between the events producing the knowledge (the actions, i.e., adding/deleting links or concept fragments, triggering the acquisition of knowledge, what Idinopulos [13] calls c*ausally mediated knowledge*) and the inferential process at its basis, i.e., how to construct the ìnewî evidence the information that is presented to the user may lead to once clicking on a link (*epistemic mediated knowledge* [13]).

From the adaptation engine point of view [24], indeed, it should not make any difference if the adaptation concerns what links to show to the user, or what text to show to the user: if the specific prerequisites are satisfied, the respective action (of adding links, or text/ multimedia content, etc.) is triggered. Actually, some systems (e.g., AHA! [8]) do not make this distinction in their adaptation model totally explicit.

We argue however that, for an adaptive hypermedia author, it is difficult to separate the two notions (links versus concepts; adaptive navigation versus adaptive presentation) and at the same time to carefully design the whole system so that adaptive navigation support actions, triggered directly by the adaptive engine, and adaptive presentation actions, triggered by in-page (or in-concept/ content) rules, are synchronous.

In the following, we propose a better way to look at the whole authoring problem in adaptive hypermedia. This approach consists of a combination of the concept mapping paradigm to construct the course narrative and of several new adaptation rules. We also show how the two formalisms may be integrated on an example version of AHA!.

2 The AHA! system

AHA! [1] is a well-known system, one of the pioneers of adaptive hypermedia (with its first version developed in 1996/97), which became almost a benchmark for the domain. One of the co-authors was involved in the research and development from the very beginnings [7], while the other is involved in supervising the project towards new developments since the support received from the NL foundation.

The power and popularity of AHA! lies on the fact that it is very simple. However, this simplicity can have drawbacks, as shown in Figure 1: if the system complexity is low, the authoring efficiency cannot be very high ñ unless the author puts a great effort into creative authoring. This is about the point where the AHA! system is now.

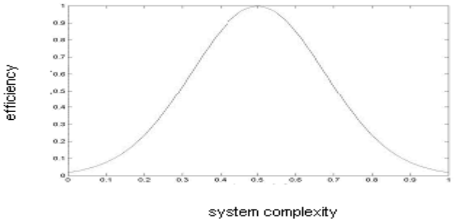

Fig. 1. Relation between system complexity and authoring efficiency

In its present form, AHA! offers the following adaptation methods:

1. each page is considered a concept, and is presented or not according to some conditions (on some variables) present in an XML file called ìrequirement listî;
2. the variables changing rules are relatively simple, and are recorded in another XML file called ìgenerate listî;
3. AHA! uses an XML based AHA tag language for conditional fragments within pages.

Items 1-2 permit adaptive navigation support at page level, and item 3 permits adaptive presentation. Recently, AHA! was extended with the following authoring tools:

1. an editor to connect requirements to pages;
2. an editor for the generate rules;
3. forms to make changes to the user model. The most important one is a form that allows the adaptive hypermedia user to modify knowledge attributes associated to page-concepts.

All these features unfortunately rely on an inadequate definition of knowledge and of knowledge acquisition because they do not seem to take into account the importance of context in concept meaning attribution (as discussed in Section 3). In particular, the possibility of altering knowledge attributes in the user model seems dangerous if this is not coupled with a redefinition of knowledge within the whole system, with the inevitable consequences that such an action has in determining concept presentation. AHA! now is moving towards database-base multiple-attribute concepts (while still trying to keep complete compatibility with the so fashionable XML format). Next, we will present some suggestions on the conceptual structure and on adaptation techniques, also pointing to possible problems that AHA! will have to face and deal with.

3 The concept-mapping paradigm

A quite intuitive classification is to divide the source material into concepts [6], as derived from the concept mapping paradigm [18]. In such a structure, each piece has an independent semantics - in the sense of the semantic Web [21]: starting with low level, *atomic concepts*, to collections of concepts (*composite concepts*), that together form a *concept hierarchy*. Concepts can relate at any level of the hierarchy. The creation of these building bricks is the role of the adaptive hypermedia designer [6].

This hierarchy represents the primitive building blocks of the hypermedia. Putting these building blocks together with different sequences generates different presentations at a relatively high granularity level (concept level). At this level, indeed, we would be only speaking of adaptive navigation support. Normally, adaptive presentation is at a lower, concepts fractions level. A simple example is the construction of text introduction. This construct can be used together with other introductory fragments in an introductory chapter, or dropped at later browsing, etc. However, such a construct has usually no independent meaning. A common solution to this is to divide concepts into sub-concepts, without sometimes caring about the loss of semantics. Such sub-concepts cannot be easily further used (in the context of *collaborative authoring*), because they cannot be semantically annotated, and therefore will not be significant for searching mechanisms.

A more appropriate solution, introduced in [6], is to sub-divide the concept into its *attributes*. These can be a concept name, alternative contents, fragments, etc. By mapping the course content on a concept hierarchy, and describing the concepts with a set of attributes, the adaptation has only to deal with *concept-level adaptation* and *attribute adaptation*. The advantage is that it can all be performed (and viewed) from a high level, and does not need separate consideration of conditional fragments written within the text, which are more difficult to re-use by other authors. In this way, the content authoring and the adaptative engine rules authoring is clearly separated, making also automatic checks easier. Adaptation is here only a matter of combining concept attributes into pages (pieces of information that can be show at a time). Navigation, in this context, is dependent on the *presentation format* (Figure 2). (e.g., a handheld device with shorter pages (than the regular browser) will display the ìnextî button more often within the same lesson(check SMIL [22], for presentation issues).

Such a model is compatible with the RDF [20] standard, where the RDF *resources* become concepts, the *properties* attributes and the *literals* attribute values. AHA! has partially implemented a similar structure, the main important difference being that concepts are at the granulation of pages, and can have a single attribute in the current implementation. Constructs within pages (such as conditional fragments) are not concept attributes in AHA!, and cannot be independently used with other concepts or concept attributes. New under development versions of AHA! consider multiple attributes, and a database structure, that will allow more flexibility ñ including hopefully dropping of the artificial separation of conditional fragments (such as in adaptive presentation) and concept linking (such as in adaptive navigation support.

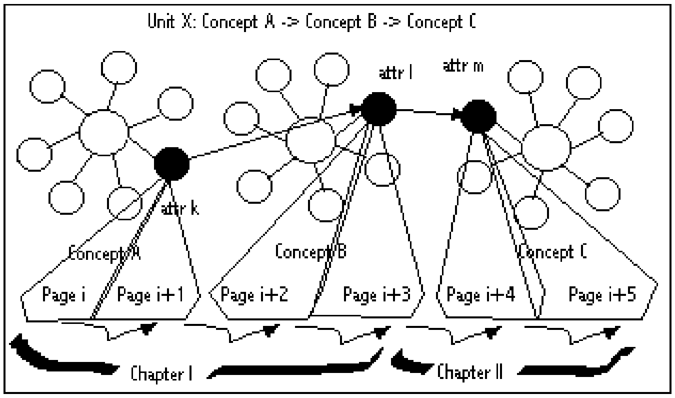

Fig. 2. A unit X is composed of, e.g., parts of 3 concepts (A, B and C) which have some attributes (attr-k, attr-l, attr-m). The presentation order is represented by directed connections between concept attributes. The unit is formed of 2 chapters that contain (parts of) the concepts. The information in a chapter is presented (by a browser) in pages (which may be shorter than a chapter). ìNextî buttons at page level are navigation support of presentation nature, and have nothing to do with the user-model related adaptation.

Figure 2 shows how connections are done at concept attribute level. Concept attributes can appear more than once, within one or more unit(s), and there is no restriction for attribute contents: text, video, audio, flash technology, etc. Next, we will explain the benefits of this simplification by presenting new adaptation types based on it.

4 New Adaptation Rules: How to Augment the Adaptation Engine

Different rules can be conceived to augment the functionality of the adaptation engine. As they trigger when some quality measurement is reached, these rules, although apparently retraceable to standard commands in traditional programming languages, can be also viewed as deriving from the Genetic Graph modeled by Goldstein [14]: both paradigms indeed explain how the userís knowledge can evolve during learning.

Most adaptive systems are rule-based. Adaptation is mainly triggered by *conditional* rules, which take the form:

```
IF <PREREQUISITE> THEN <ACTION>
```

We propose some alternative forms of conditional rules that allow more freedom, both in authoring as well as in studying with this type of environment.

1. A *level* rule [6]:

   IF ENOUGH (<PREREQUISITES>) THEN <ACTION>

   where ENOUGH = function of number and quality of prerequisites; true when, e.g., a given number of prerequisites from a PREREQUISITES set is fulfilled. This type of relaxation of the prerequisites is intuitive, in the sense that it allows the author to write simplified rules, instead of writing a great number of complex ones; the idea is derived from game levels.

   Example: PREREQUISITES = time_spent; ACTION = ìgo to next levelî

   ```
   The rule becomes:
       IF ENOUGH (time_was spent on current level) THEN ìgo to next levelî
   Where ENOUGH is defined, e.g., as follows:
       ENOUGH (time) = 30 time units;
       time (advanced topic) = 10 (time units per topic);
       ENOUGH (medium topic) = 5 (time units per topic);
       ENOUGH (beginner topic) = 2 (time units per topic);
   ```

2. A *temporal* rule: a certain action is repeated as long as one or more conditions hold. To capture unbound minimization, we add therefore the WHILE construct:

   WHILE <CONDITION> DO <ACTION>

   According to the concept mapping paradigm, concepts are not canned but are assembled depending on the user model based on their attributes (see Section 3). Therefore this operation requires more than the mere addition/deletion of links.

3. A *repetition* rule: a certain action (simple / composed) is repeated for a number of times that is predefined by the author. This rule forces the user to reread a concept (as described in [15]) that is presented to her invariably. In non-educational contexts, this rule might have a lyrical effect[1] ñ such as a refrain in a song.

   The temporal rule could be expressed as:

   FOR <i=1…n> DO <ACTION>

   This rule therefore describes the time this action has to last before the reader can move on to another one.

4. An *interruption* command: the action the user is performing is interrupted (broken) and she is forced to undertake a different one abruptly.

   BREAK <ACTION>

The adoption of this rule might be due to lyrical reasons. This is the case in *A life set for two* [11], where the reader is forced to reach the end of the fiction once she has read a predefined percentage of it In educational contexts, this rule might be used by the author to constraint the userís explorative behavior within predefined boundaries, i.e., those corresponding to the pedagogical rules implemented in the system. In this sense, the implementation of this rule represents an exacerbation of the traditional behavior of AH systems: here, indeed, the user is ìpunishedî if she does not stick to the learning pathways provided by the system.

5. A *generalization* command: the new concept the reader has come across is compared with the more general ones it refers to. As a result of this inductive action, the reader is pointed to the related concepts she may be interested in reading. So, the reader has selected a node that describes a certain concept in specific, individ-

[1] The lyrical use of repetitions in hyperfiction has given rise to a particular design pattern as described in [2].

ual terms. The system ì interpretsî this behavior as an interest from the reader about that particular notion and therefore performs an inductive action to point her to the more general ones it refers to.

GENERALIZE (COND, COND$_1$, Ö , COND$_n$)

6. A *specialization* command: conversely, if the concept is general, the system deductively points the reader to its more specific instantiations.

SPECIALIZE (COND, COND$_1$, Ö , COND$_n$)

So, for instance, if a student is reading a page about the concept of the ì Model Readerî in a course about postmodern literature, she could be further pointed to an extract from Calvinoís novel ëSe una notteí, where this notion is exemplified, or to a further theoretical elaboration on the same topic (by, for example, Genette), or, again, to a description of how this idea is realized in hyperfiction (see [4]).

Other commands that could appear are *comparison* (concept analogy search) and *difference* ñ both instances of generalization; or *duration* ñ a rule related to repetition.

The appropriateness of these rules directly depends on the context and on the concepts to be modeled. Certainly, educational material may be simpler and more straightforward to model than a less formal one, like, for instance literary material. In an educational setting, Indeed, the assumption is often to guide the student to evolve from novice to expert in the content domain [5]: this is why the content structure has to be well defined and normally hierarchically organized. Literature, however, exemplifies a case where not only the readerís goal is not clear (contrary to the learning goal of educational systems[2]) and, as a consequence, the way readers use information to construct meaning [12], but, also, it enacts a different approach: not a ì guided pullingî approach [19] like in educational systems, but one based on a ì suspension of disbelievingî [9], on the importance of rereading in constructing meaning [15], on suspense, on playing with the reader[3] .

Some of these principles would however be effective also in educational systems: we think, for instance, of the notion of rereading (the old say ì repetita iuvantî); of suspension, of disbelieving as a way of addressing the question of authority and authenticity by assigning it to the author of the content; of a sort of suspense in the way information is presented to readers as to encourage and to motivate them to read further. Moreover, the possibility of expressing these sorts of rules is precisely what guarantees the general-purpose character of the underlying formalism.

4 Implementing New Rules in the Current AHA!

Following the present syntax used by AHA! to express rules, we could extend it to represent the above mentioned functions in the following way :

[2] For an overview of the possible goals and their related reading strategies adopted by readers in hyperfiction see, for instance, [4].

[3] We refer again to the illuminating paper in [12], where they describe one of the first systems developed to deal with adaptivity in literature.

1. :the level rule:	**2. the temporal rule:**
<if expr=îenoughî> <enough> Here definition of enough </enough> <block> Here a conditional fragment </block> <block> Here an optional alternative fragment </block>	<while expr=ìwrong_search&gt;50î> <block> Here a conditional fragment </block> </while> In this case, a certain action is repeated as long as a certain condition holds. For instance, a warning is repeated that the search performed by the user is in the wrong direction, while the system is still performing the search. Another condition then could trigger a service denial response if the above threshold would be passed.
	Another example is: <WHILE expr=ìart&gt;70 and not culture&gt;80î> <BLOCK> Here a predefined sequence of events </BLOCK> <BLOCK> Here an alternative event (action) </BLOCK> </WHILE>
3. the repetition rule:	**4. the interruption command:**
<for expr=îquestion&sm;5î> <block> Here a conditional fragment </block> </for> In a particular context, this action may make sense. For the example above, some explanation is given as long as the number of questions is not larger than 5. After that, a different strategy has to be taken into consideration ñ such as the suggestion of consulting different material, etc.	<break> <block> Here a conditional fragment </block> </break>
5.the generalization command:	**6.the specialization command:**
<generalize concept=îmyconceptî> Here details of generalization (levels, etc.) </generalize> Example: <generalize concept= ìdouble_code_theoryî> Here details of generalization (levels, etc.) </generalize> Such a rule can be used to jump 1 or more levels in the hierarchy of concepts. Extra processing can be done in the body of the above command, such as giving comments on the level to be visited and the reason why.	<specialize concept=îmyconceptî> Here details of specialization (levels, etc.) </specialize> Example: <specialize concept=ìModel Readerî> Here details of specialization (levels, etc.) </specialize> The application is similar as the above command, with the difference that the direction of the processing in the concept hierarchy is *top-down* instead of *bottom-up*.

4 Problems and the need of checking mechanisms

As shown in Figure 1, by increasing the system complexity, the authoring efficiency grows for a while, and then drops. The problem with the current AHA! system is that is it somewhere at the beginning of the slope. Adding to many more features and flexibilities can increase the authoring efficiency for a while, but it is necessary to stop before the down-curve. When authors have to deal with complex unit graphs with many concepts and many concept attributes, it is easy to leave something out by mistake. AHAM [24] tries to deal with such problems as:

- *termination* (avoiding of loops) and
- *confluence* (equivalence of order of rule execution - for rule-based adaptation engines).

Their suggestion is to use *activation graphs* (from the static analysis of active database rules), therefore constructing the whole graph of possible states that is determined by the concepts, their links, the attributes, their values (especially, initial values and possible ranges ñ to eliminate unnecessary branches and optimize the search tree with the help of constrains) and the rule sets. If such a graph has no loops, the system will always terminate. For confluence, a difficult procedure of checking the possibility of *commutation* between each rule pair (so their order equivalence) is proposed. AHA! at the moment ensures termination by allowing only monotonic increases of the attributes (per concept), which will be more difficult for the following version with multiple attributes. As for confluence, AHA! doesnít deal with it at all.

Other problems that can appear are:

- concepts (or concept fragments) *never reached*;
- rules (or other adaptation mechanisms) that generate attributes with *out of range (or domain) values.*

The good news is that the added rules that we have proposed donít require extra checking mechanisms than the ones studied previously in the AHAM context [24]. Basically, loops existing in the regular rules will also be present (and noticeable) in level rules, temporal rules or repetition rules. (and vice-versa). Non-equivalent rules that can be executed at a given time but are not commutable will pose the same problems on regular rules as on the extended set. The extended commands of generalization and specialization can be treated the same as regular links (or rules). The interruption command can actually help in breaking infinite loops, or signaling problems, similar to the *catch-throw* mechanism of exception handling in Java.

The bad news is that such a mechanism can be rather *time* - and *space* -consuming.

A better way of dealing with this problem is by means of various simplifications and complexity decreasing assumptions. One such simplification could be by means of a belief revision technique to check inconsistencies among knowledge attributes to concepts and the consequent knowledge acquisition problem. Belief revision consists in the introduction of a sort of case-based heuristics that:

1. recalls a previous concept with the same features and its associated attributes;
2. adapts, via some rule-based formalism, the course structure (the narrative) to the current learning scenario;
3. resolves the emerging inconsistencies to make sure that changes of state are epistemologically conservative (so that the resulting narrative is not subverted).

6 Future directions

With the standardization of the building bricks of adaptive systems (such as LOM [17], Learner model [16]ñ IEEE, LTSC for education, RDF [20], etc.) it becomes more and more feasible to collaborate and share adaptive techniques, technologies but also, system parts, developed adaptive hypermedia presentations, etc. Adaptive and adaptable systems are necessary in education, where learners come with different cultural and knowledge backgrounds, different learning styles, genders, ages, especially in the new context of life-long learning. Such systems are definitely necessary in commerce (and are having a tremendous success in e-commerce, even with extremely simple adaptation techniques ñ such as the well-known Amazon.com technique of suggesting ì buyers who bought this book also bought Ö .î). But they can have, as we have shown also some other, surprising applications, such as adaptive literature and adaptive art.

It is extremely important to find the right balance between system complexity and authoring efficiency, as shown in figure 1. Extending adaptive systems with extra adaptation rules can be beneficial, especially if these rules can express situations that were not possible (or difficult) to express with the given set of tools/ rules. Moreover, this enhancement makes sense if it does not impose on the checking mechanisms, by increasing dramatically the types of tests an adaptive hypermedia author has to perform in order to verify his/her output.

As a direction we predict to be rewarding for such checking mechanisms we see the replacement of large state trees with all possible situations to be reached from the existing rule base (or, generally speaking, adaptation procedure) with a visual, dynamical representation of the processes involved. For example, the effect of a new rule on the rest can be shown on the static (and much smaller) unit link graph, as the propagation of some colored fluid through the graph, etc.

7 Conclusion

The paper has started with a criticism on the widespread practice to distinguish adaptation in hypermedia between an adaptive navigation support and an adaptive presentation. This criticism is based on the claim that, in this way, authors of adaptive courseware have to artificially separate links from concepts but still to coordinate and tune them in order to provide an adaptation that is conceptually valid and that contributes to a significant knowledge acquisition.

We have suggested a better way to look at the whole authoring problem in adaptive hypermedia. This approach consists in the combination of the concept mapping paradigm to construct the course narrative and of several new adaptation rules. We have highlighted a few new rules that should be integrated into an adaptive authoring shell or toolkit.

We have shown how these two formalisms may be integrated in an example version of AHA!. Moreover, we have augmented the present rule behavior performed by AHA!

to allow it more adaptivity. We claim that this approach is another step towards flexible generic-purpose adaptive hypermedia.

References

1. AHA, Adaptive Hypermedia for All; http://aha.win.tue.nl/
2. Bernstein, M.: Patterns of Hypertext. Ninth ACM Conference on Hypertext (1998) 21-29
3. Brusilovsky, P.: Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction 6 (1996) 87-129
4. Calvi, L.: ëLector in rebusí: the Role of the Reader and the Characteristics of Hyperreading. Tenth ACM Conference on Hypertext í99. ACM Press (1999) 101-109
5. Calvi, L., De Bra, P.: Using Dynamic Hypertext to Create Multi-Purpose Textbooks. In T. M¸ldner and T.C. Reeves (eds.), ED-MEDIAí97, AACE Press (1997), 130-135
6. Cristea, A., Okamoto, T.: MyEnglishTeacher ñ A WWW System for Academic English Teaching ICCE 2000, Taipei, Taiwan (2000)
7. De Bra, P., Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia (1998) 115-139
8. De Bra, P., Ruiter, J.P.: AHA! Adaptive Hypermedia for All. WebNet (2001) 262-268
9. Eco, U.: Six Walks in the Fictional Woods. Harvard University Press (1994)
10. Goldman, A.: Telerobotic Knowledge: A Reliabilist Approach. In K. Goldberg (ed.). The Robot in the Garden. Telerobotics and Telepistemology in the Age of the Internet. MIT Press (2000) 126-142
11. Kendall, R.: A Life Set For Two. Eastgate Systems, Watertown, MA (1996)
12. Kendall, R., RÈty, JH.: Toward an Organic Hypertext. Hypertext í00. (2000) 161-170
13. Idinopulos, M.: Telepistemology, Mediation, and the Design of Transparent Interfaces. In K. Goldberg (ed.). The Robot in the Garden. Telerobotics and Telepistemology in the Age of the Internet. MIT Press (2000) 312-329
14. Goldstein, I.: The genetic graph: a representation for the evolution of procedural knowledge. In D. Sleeman and J.S. Brown (eds, Intelligent Tutoring Systems. Academic Press, 1982.
15. Joyce, M.: Nonce Upon Some Times: Rereading Hypertext Fictions. Modern Fiction Studies 43(3) (1997) 579-597
16. Learner Model. IEEE Learning Technology Standards Committee (LTSC); http://ltsc.ieee.org/wg2/index.html
17. LOM (Learner Object Metadata). IEEE Learning Technology Standards Committee (LTSC); http://ltsc.ieee.org/wg12/index.html
18. Novak, J. D.: Clarify with concept maps. The Science Teacher, 58(7), (1991) 45-49
19. Perrin, M.: What Cognitive Science Tells us About the Use of New Technologies. Invited Talk given at the Conference on Languages for Specific Purposes and Academic Purposes: Integrating Theory into Practice. Dublin, 6-8 March (1998)
20. RDF (Resource description framework) standard: http://www.w3.org/RDF/
21. Semantic Web: http://www.w3.org/2001/sw/
22. SMIL, W3C standard, http://www.w3.org/AudioVideo/
23. W3C Recommendation: Authoring Tool Accessibility Guidelines 1.0 (3 February 2000); http://www.w3.org/TR/ATAG10/
24. Wu, H., De Bra, P.: Suficient Conditions for Well-behaved Adaptive Hypermedia Systems. Web Intelligence. Lecture Notes in AI. Vol. 2198. Springer. (2001) 148-152