Authoring Multimedia Documents using WYSIWYM Editing

Identification number: 718

Abstract

This paper does three things:

(1) It outlines a future 'ideal' multimedia document authoring system that allows authors to specify content and form of the document independently of each other and at a high level of abstraction;
(2) It describes a working system that implements a small but significant part of the functionality of such an ideal system, based on semantic modeling of the pictures as well as the text of the document; and
(3) It explains what needs to be done to bridge the gap between the implemented system and the ideal one.

1 A Future Ideal Multimedia Document Authoring System

A *Document Authoring System* is a tool that helps an author to write documents. If the system supports the authoring of documents that combine 'presentations' in different media (text and images, for example), we will speak of a *multimedia* document authoring system. *Ideally*, a multimedia document authoring system would allow authors to specify the content and form of a high-quality document in ways that are both simple and efficient. More specifically, an ideal system would afford the following options to the author:

1. *Easy determination of content.* 'Content' is taken to mean the factual (i.e., propositional) content of the document – in other words, the content of the Knowledge Base (KB) that forms the input to the authoring system.

2. *Easy determination of style and layout.* In the absence of specific instructions from the author, style and layout should be determined using intelligent defaults. (For example, the standard settings may require the document to be informal, with avoidance of technical terms, lists and footnotes, without maximum paragraph length, and with numbered sections.) Defaults can be overridden by the author, whereupon other defaults may become relevant.

3. *Easy allocation of media.* As in the case of style and layout, the system has to use judiciously chosen defaults: perhaps using illustrative pictures wherever suitable pictures are available, and graphs wherever quantitative information (of a certain kind) is involved. As above, defaults may be overruled by specific requests from the author; if a request is impossible to fullfil, an appropriate error message should be generated.

4. *Easy annotation of non-generated presentations.* In some cases, it will be possible for the system to *generate* presentations. In other cases, this may be impossible: Literally quoted texts, for example, or historic photographs, may predate the use of the system, in which case it may be necessary to treat them as 'canned' and to annotate them to allow the system to make intelligent use of them.

5. *Easy post-editing.* Once the system has produced a document according to the specifications of the author, the ideal system would offer tools to address remaining inadequacies using post-editing.

'Easy' means efficient, protected against inconsistencies, and not requiring specialist skills or knowledge. A domain specialist – who may not know anything about knowledge representation, logic, or linguistics – could use such a system to build KBs that the system can turn into documents in any desired language using any desired combination of media. The production and updating of complex documents would be greatly simplified as a result.

In present-day practice, these requirements tend to be far from realized: authoring documents by means of such tools as MS WORD or POWERPOINT requires much low-level interaction, such as the typing of characters on a keyboard and the dragging of figures from one physical location to another. In some cases, an Intelligent Multimedia Presentation System (IMMPs e.g., Bordegoni et al. 1997) can be used (see AIR 1995, Maybury and Wahlster 1998 for some surveys), which employs techniques from Artificial Intelligence to allow higher-level interaction. Present IMMPs, however, meet few of the requirements mentioned above. Most of them, for example, require input of a highly specialized nature (e.g., the complex logical formulas entered in the WIP system, André and Rist 1995)[1] and they allow an author little control over the form (e.g., layout, textual style, media allocation) of the document. The issue of easy annotation (4) is never even addressed, to the best of our knowledge.

The next section describes an implemented system for the authoring of *textual* documents that can be argued to fulfill requirements (1) and (2) and which forms a suitable starting point for working towards the 'ideal' *multimedia* system outlined above. Section 3 describes an extension of this system in which significant aspects of requirements 3-5 have also been implemented. Key features of this new system are its ability to use *semantic representations* that are common to the different media, and the ability to construct natural language *feedback texts* to help the author understand the content and the form of the document while it is still under construction. The concluding section explains what needs to be done to fill the gap between the implemented system and the ideal one.

[1] An exception is ALFRESCO which takes natural language input, requiring the system to interpret unconstrained natural language (Stock 1991). Avoiding the need for doing this is an important design motivation for WYSIWYM-based systems such as the ones described in sections 2 and 3.

2 A WYSIWYM-based System for the Authoring of Textual Documents

Elsewhere (Power and Scott 1998, Scott et al. 1998, Scott 1999), a new knowledge-editing method called 'WYSIWYM editing' has been introduced and motivated. WYSIWYM editing allows a domain expert to edit a knowledge base (KB) by interacting with a *feedback text*, generated by the system, which presents both the knowledge already defined and the options for extending and modifying it. Knowledge is added or modified by menu-based choices which directly affect the knowledge base; the result is displayed to the author by means of an automatically generated feedback text: thus 'What You See Is What You Meant'. WYSIWYM instantiates a general recent trend in dialogue systems towards moving some of the *initiative* from the user to the system, allowing such systems to avoid the difficulties of processing 'open' (i.e., unconstrained) input.

Of particular importance, here, are applications of WYSIWYM to the generation of documents containing text and pictures; the present section focuses on (multilingual) *text* generation: the KB created with the help of WYSIWYM is used as input to a natural language generation (NLG) program, producing as output a document of some sort, for the benefit of an end user. Present applications of WYSIWYM to text generation use a KL-ONE-type knowledge representation language as input to two NLG systems. One NLG system generates feedback texts (for the author) and the other generates output texts (for an end user). One application currently under development has the creation of Patient Information Leaflets (PILLs) as its domain. The present version of this PILLs system allows authors to enter information about possible side effects (*'If you are either pregnant or allergic to penicillin, then tell your doctor'*) and how to handle medical devices such as inhalers, inoculators, etc. By interacting with the feedback texts generated by the system, the author

can define a procedure for performing a task, e.g. preparing an inhaler for use. A new KB leads to the creation of a procedure instance, e.g. p. The permanent part of the KB (i.e., the T-Box) specifies that procedures may be complex or atomic, and lists a number of options in both cases. In the atomic case, the options include `Clean, Store, Remove`, etc., and these are made visible by means of a menu from which the author can select, say, `Remove`. The program responds by adding a new instance, of type `Remove`, to the KB:

Remove(p)

('There is a procedure p whose type is `Remove`.') From the updated knowledge base, the generator produces a feedback text

> Remove **this device or device-part** from **this device or device-part**,

making use of the information, in the T-Box of the system, that `Remove` procedures require an Actee and a Source. Such not yet defined attributes are shown through mouse-sensitive anchors. By clicking on an anchor, the author obtains a pop-up menu listing the permissible values of the attribute; by selecting one of these options, the author updates the knowledge base. Clicking on **this device or device part** yields a pop-up menu that lists all the types of devices and their parts that the system knows about, including a Cover (which, according to the T-Box must have a Device as Owner). By continuing to make choices at anchors, the author might expand the knowledge base in the following sequence:

- Remove **a device**'s cover from **a device or device-part**
- Remove **a device**'s cover from an inhaler of **a person**
- Remove **a device**'s cover from your inhaler
- Remove your inhaler's cover from your inhaler

At this point the knowledge base is potentially complete, so a (less stilted) *output text* can be generated and incorporated into the leaflet, e.g.

> Please remove the cover of your inhaler.

Longer output texts can be obtained by expanding the feedback text further. A number of properties of the PILLs system are worth stressing. First, the system supports a high-level dialogue, allowing the author to disregard low-level details, such as the exact words used in the output text. This makes it possible to interact with the system using, say, French (provided a generator for French *feedback* texts is available), for the production of leaflets in Japanese (provided a generator for Japanese *output* texts is available). The semantic model in the T-Box guarantees that many types of inconsistencies (e.g., a medicine that has to be taken both once and twice a day) are prevented. Second, a simple version of WYSIWYM has also been applied to the form of the text, allowing the author to specify it separately from its content. This is done by allowing the author to use WYSIWYM for building a second, form-related KB which describes the *style and layout* of the document. This KB, for example, may state that the maximum paragraph length is 10 sentences and that there are no footnotes. (A second, form-related T-Box determines what the options determining layout are.) This form-related KB constrains the texts that are generated. By interacting with feedback texts describing the form-related KB, the author changes the stylistic/layout properties of the document.

3 A WYSIWYM-based System for the Authoring of Multimedia Documents

ILLUSTRATE is an extension of PILLs producing documents that contain pictures as well as words. Consider a toy example, adapted from ABPI (1997). Suppose the document says *Remove the cover of your inhaler.* An instruction of this kind may be illustrated by the picture below. How can a document authoring system produce a document in which appropriate pictures illustrate the text when this is desired? ILLUSTRATE does this by allowing an author to ask for pictorial illustration of the information in the document by interacting with the feedback texts. The author can indicate, for a given mouse-sensitive stretch s of the feedback text, whether she would like to see the part of the

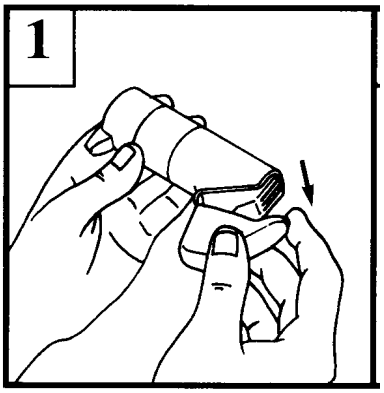

Figure 1: One of the pictures in the library of the authoring system

document that corresponds to s illustrated.[2] If so, the system searches its library to find a picture that matches the meaning of s. In Fig.2, the author has requested illustration of the instruction corresponding with the text 'Remove your inhaler's cover from your inhaler'. (The other four options are irrelevant for present purposes.) In domains where all the pictures are

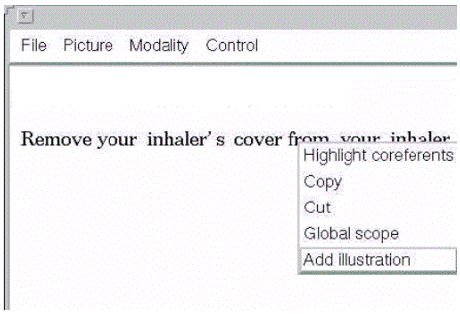

Figure 2: Screendump: Author makes a requests for illustration

variations on a common theme, suitable pictures can be *generated*. In the case of Patient Information Leaflets, however, this was not a practical option because of the many different kinds of things depicted in the leaflets: medicine packages, body parts, medical applicances, various types of actions, etc. Pictures, moreover, are heavily reused in the different leaflets written by the same company. For these reasons, ILLUSTRATE uses an alternative approach, *selecting* pictures from a library, each of which is annotated with a formal representation of its meaning. We will explain the workings of ILLUSTRATE by answering three questions: (1) What kinds of representations are used in the library to annotate the pictures with relevant aspects of their meaning? (2) How is the semantically annotated library of pictures created? and (3) What selection algorithm is employed to retrieve an optimally appropriate illustration for a given part of the KB from the library? We shall assume that the information whose illustration is requested corresponds with the following formula in the KB, which represents the meaning of the feedback text in Fig. 2.

$$\begin{aligned} &Remove(p) \;\&\; Actor(p) = \\ &x \;\&\; Reader(x) \;\&\; Source(p) = y \;\& \\ &Inhaler(y) \;\&\; Actee(p) = z \\ &\;\&\; Cover(z) \;\&\; Owner(z) = y. \end{aligned}$$

('There exists a 'Remove' action whose Source is an Inhaler and whose Actee is a Cover of the same inhaler.')

1. What kinds of representations are used? Representations say what information each picture *intends to convey*. Irrelevant details should be omitted. It has been observed that photographic pictures express 'vivid' information and that this information can be expressed by a conjunction of positive literals (Levesque 1986). In line with this observation, ILLUSTRATE represents the meaning of the picture in Fig. 1, for example, as follows:

$$\begin{aligned} &Remove(p) \;\&\; Source(p) = \\ &y \;\&\; Haler(y) \;\&\; Actee(p) = z \\ &\;\&\; Cover(z) \;\&\; Owner(z) = y. \end{aligned}$$

If any of the variables e, x, y, z has an occurrence in the meaning representation of another picture then these occurrences corefer. This allows the system to know when two pictures depict the same person, for example (REF xxx).

2. How is the library created? This is a question of great importance because the library contains semantic representations that are much more detailed than those in current picture retrieval systems (e.g. Van de Waal 1995) and this

[2] In fact, ILLUSTRATE also allows authors to express requests for illustration by highlighting a formula in the KB (instead of a piece of feedback text), thus accommodating technically more expert authors.

could potentially make the annotation task extremely burdensome (Enser 1995). The answer to this problem may be unexpected: ILLUSTRATE uses WYSIWYM itself to enable authors to associate a given picture with a novel representation. The class of representations that are suitable for expressing the meaning of a picture is, after all, a ('vivid') subset of the class of representations allowed by the T-Box for the text of the document, and consequently, the same WYSIWYM interface can be used to create such representations. Fig. 3 contains a screendump of the annotation process, where the current annotation corresponds with the formula $Remove(p)$ & $Source(p) = y$ & $Actee(p) = z$ & $Cover(z)$ & $Owner(z) = y$. Note that this formula is still incomplete because the nature of the Source is undefined. (When it is finished, the feedback text will be equivalent to that in Figure 1.) The top of the screendump shows the accompanying feedback text containing anchors for further additions.

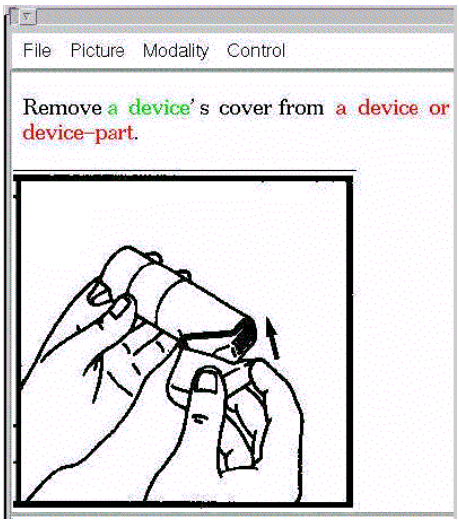

Figure 3: Screendump: A stage during the annotation of a picture

3. What is the selection algorithm? Notice that a picture can illustrate an item of information without expressing all the information in it. For example, Fig. 1 does not show that the Actor is the Reader and it leaves the type of 'Haler' unspecified. (The leaflets describe *In*halers, *Autohalers*, and *Aerohalers*.) Therefore, a selection rule must allow pictures to omit information:

> **Selection Rule:** *Use the logically* **strongest** *picture whose representation is logically* **implied by** *the information to be illustrated.* [REF xxx]

Logical strength is determined on the basis of the two semantic representations alone. Determining whether one representation logically implies the other, where one is an instance in the KB and the other a representation of a picture, is easy, given that both are conjunctions of positive literals [REF xxx].

This brief description should suffice to highlight the following advantages of ILLUSTRATE:

- One uniform interface is employed for all actions that involve the editing of semantic representations, regardless of the type of presentation involved (i.e., its media).
- When used for the construction of annotations of pictures, the T-Box of the system makes sure that only those properties can enter an annotation that are relevant in connection with it. For example, the height of the patient is usually irrelevant, and consequently the T-Box does not make height an attribute of a person.
- Pictures are retrieved by a reasoning process that involves classical logic; since a match between a picture and a piece of the KB can never be inexact, there is no need to complicate the retrieval process by making it probabilistic, as has to be done when the system has less control over the form of annotations (Van Rijsbergen 1985, REF[xxx]).

Specific aspects of ILLUSTRATE have been described elsewhere, but the assumptions behind the system as a whole have not been stated before. (For the representation scheme and the selection scheme see [REF xxx]; for the treatment of *sequences* of pictures see [REF xxx].) We have so far simplified by assuming there to be only one author. In fact, however, an intelligent authoring system is most useful when there are several authors (each of which can be allowed to work in a different language). More specifically, it is plausible that the person authoring the annotations in the library is not the

same as the person(s) who author(s) the document itself.

4 Gap Analysis: Future Work Towards the Ideal

The PILLs system (section 2) makes a fair stab at fulfilling text-related requirements 1 and 2 mentioned in section 1. The ILLUSTRATE demonstrator goes beyond this, fulfilling important aspects of requirements 3 and 4 as well. Yet, there is a considerable gap between the implemented system and the ideal one of section 1. Possible improvements do not only concern the coverage of the system (i.e., the types of text-picture combinations occurring in the documents) and the quality of the documents, but matters of system architecture as well. Three different sets of improvements may be discerned. Firstly, there is requirement 5 of section 1: *Easy postediting.* It is easy to allow authors to make low-level corrections in the document *after* the interaction with WYSIWYM. It would be useful, however, to allow low-level editing while retaining the connection between the edited document and the content of the various knowledge bases (the content-related KB, the form-related KB, and the representations in the library). This, however, would require the system to 'understand' the meaning of the low-level editing actions performed (e.g. changing the order of two expressions), and this seems an unreasonable requirement. We have to conclude that this form of post-editing is not possible given the state of the art in text- and picture understanding.

Some other improvements would be less problematic. On the one hand, there are issues that have been tackled by other research groups and whose solutions we intend to carry over to a WYSIWYM-based setting. These concern the generation of graphics from underlying representations (Wahlster et al. 1993) and the problem of optimizing the layout of text & picture documents (e.g. Graph et al. 1996), for instance. Three remaining improvements, on the other hand, are matters for future research:

- *Media allocation.* ILLUSTRATE embodies one way in which media may be allocated. Other mechanisms could give the system more autonomy. For example, the system may use rules (e.g. Roth and Hefley 1993) to decide autonomously what information is in need of illustration. Such rules may be used as defaults, to be overruled by authors' requests. Similarly, authors may be enabled to point at thumbnail pictures, whereupon the system tries to find a suitable place in the document to include them, based on the representation of their meaning and making use of Rule A (section 3). By thus allowing the author and the system to cooperate on media allocation, this difficult task will be made more tractable (see the recent discussions in ETAI 1997-8).
- *Other media.* Little in ILLUSTRATE hinges on the fact that the objects in the library are pictures. The same system, for example, can be used for annotating sound or *canned text* (for example, a complex bit of law code, which needs to be rendered literally). Of great practical interest, finally, is the possibility of including documents authored previously (and possibly by a different author), leading to iterative application of WYSIWYM.
- *Interaction between media.* Ideally, the words in a text should be affected by the inclusion of a picture: First, and most obviously, texts may be *enlarged* by references to pictures (e.g., references like 'See Fig. 3' may be added, cf. Paraboni and Van Deemter 1999). Secondly, texts may be *reduced* because information expressed in the picture can be shortened (or left out altogether). One type of situation where this happens is exemplified by the text 'Remove the capsule from the foil as shown in the picture' (ABPI 1997), accompanied by a picture showing how this may be done. Other types of situation include the case where quantitative information is expressed through a *vague* textual description ('a blob of cream', 'a fingertip of ointment') that is made more precise by means of a picture showing the required amount.

It should be noted that each of these extensions depends crucially on ILLUSTRATE's ability to manipulate the semantic representations associated with multimedia objects, where one and the same representation language is used for

the different media: a multimedia 'interlingua' (e.g. Barker-Plummer and Greeves 1995). In the case of an author selecting a picture using thumbnails, for example, the semantic representation enables the author to (a) find a suitable location for the picture and (b) adapt the text by omitting from it information that is now expressed by the picture.

References

ABPI. 1997. The Association of the British Pharmaceutical Industry, *1996-1997* ABPI *Compendium of Patient Information Leaflets.*

AIR. 1995. Special Issue, edited by P. Mc Kevitt, on Integration of Natural Language and Vision Processing: Intelligent Multimedia. Artificial Intelligence Review 9, Nos.2-3.

E. André and Th. Rist. 1995. Generating Coherent Presentations Employing Textual and Visual Material. *Artificial Intelligence Review* **9**:147-165.

D. Barker-Plummer and M. Greeves. 1995. Architectures for Heterogeneous Reasoning. In J.Lee (Ed.) *Proc. of First International Workshop on Intelligence and Multimodality in Multimedia Interfaces: Research and Applications* (IMMI-1), Edinburgh.

M. Bordegoni, G. Faconti, S. Feiner, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson. 1997. A Standard Reference Model for Intelligent Multimedia Presentation Systems. Computer Standards & Interfaces 18, pp. 477-496.

P. Enser. 1995. Progress in Documentation; Pictorial Information Retrieval. *Journal of Documentation*, Vol.51, No.2, pp.126-170.

ETAI (1997, 1998). ETAI News Journal on Intelligent User Interfaces, Vol 1, No's 1 and 2.

W.H. Graf, S. Neurohr, and R. Goebel. 1996. A Constraint-Based Tool for the Pagination of Yellow-Page Directories. In U. Geske and H. Simonis (Eds.) Procs. of KI96 workshop on declarative constraint programming. GMD-Studien 297, St. Augustin.

H.J. Levesque. 1986. Making Believers out of Computers. Artificial Intelligence 30, pp.81-108

M. Maybury and W. Wahlster. 1998. Readings in Intelligent User Interfaces. Morgan Kaufmann Publ., San Francisco.

I. Paraboni and K. van Deemter. 1999. Issues for Generation of Document Deixis. In E. André et al. (Eds) *Procs. of workshop on Deixis, Demonstration and Deictic Belief in Multimedia Contexts*, in association with the 11th European Summers School in Logic, Language and Information (ESSLLI99).

R. Power and D. Scott. 1998. Multilingual Authoring using Feedback Texts. In *Proc. of COLING/ACL conference*, Montreal.

S. Roth and W. Hefley. 1993. Intelligent Multimedia Presentation Systems: Research and Principles. In M.Maybury (Ed.) *Intelligent Multimedia Interfaces*, AAAI Press, pp.13-58.

D. Scott, R. Power, and R. Evans. 1998. "Generation as a Solution to its own Problem", Accepted for Proc. of 9th International Workshop on Natural Language Generation, Aug.1998.

Scott, D. 1999. The Multilingual Generation Game: authoring fluent texts in unfamiliar languages. Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99).

O. Stock. 1991. Natural Language and Exploration of an Information Space: the ALFfresco Interactive System. In M. Maybury and W. Wahlster (1998).

H. van de Waal. 1995. ICONCLASS; *An iconographic classification system.* Amsterdam 1973-1985 (17 vols). ISBN 0-7204-8264-X. See also `<http://iconclass.let.ruu.nl/home.html>`.

C.J. van Rijsbergen. 1989. Towards an information logic. In: Proc. ACM SIGIR.

W. Wahlster, E. André, W. Finkler, H.-J. Profitlich, and Th.Rist. 1993. Plan-based Integration of Natural Language and Graphics Generation. *Artificial Intelligence* 63, p.387-427.