Multimedia Document Authoring using WYSIWYM editing

Kees van Deemter and Richard Power
ITRI, University of Brighton,
Lewes Road, Brighton, BN2 4GJ, United Kingdom

Kees.van.Deemter@itri.brighton.ac.uk
Richard.Power@itri.brighton.ac.uk

Abstract

This paper[1] outlines a future 'ideal' multimedia document authoring system that would allow authors to specify content and form of the document independently of each other and at a high level of abstraction. One of the main challenges in a system of this kind is to ensure the *coherence* of the generated document. This implies, among other things, that the information expressed by the different media used by the document complements each other optimally and that readers grasp the connections between them. Having introduced the 'ideal' system, we describe a working system that implements a small but significant part of the functionality of such a system, based on semantic modeling of the pictures as well as the text of the document. Finally, we sketch what needs to be done to bridge the still considerable gap between the implemented system and the ideal one.

1 A Future Ideal Multimedia Document Authoring System

Document authoring systems based on symbolic authoring (e.g., Power and Scott 1998) allow authors to create a knowledge base (KB) which is turned into a textual document by a Natural Language Generation (NLG) program. The present paper discusses an extension of this paradigm that supports the authoring of *multimedia* documents. Ideally, this involves

1. *Easy determination of content.* The system would make it easy for the author to determine the factual (i.e., propositional) content of the Knowledge Base (KB) that forms the input to the authoring system. The system would guarantee that the document generated is faithful to the content of the KB.
2. *Easy determination of style and layout.* Style and layout of the target document would be determined using intelligent defaults which can be overridden by the author if necessary.
3. *Easy allocation of media.* The system would use judiciously chosen defaults for the allocation of media: perhaps using illustrative pictures wherever suitable pictures are available, and graphs wherever quantitative information is involved.
4. *Easy annotation of non-generated presentations.* In some cases, it will not be possible for the system to *generate* presentations. 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 before the system can make intelligent use of them.

Producing and updating multimedia documents would be greatly simplified if an 'ideal' system of this kind existed: it would allow a domain specialist (who may not know anything about logic or linguistics) to compose coherent documents, in which the different media complement each other optimally. In present-day practice, requirements 1-4 tend to be far from realized: authoring documents by means of such tools as POWERPOINT requires much low-level interaction (e.g. the typing of characters or the dragging of figures from one physical location to another) whereas most Intelligent Multimedia Presentation System (IMMPS e.g., Bordegoni et al. 1997, AIR 1995, Maybury and Wahlster 1998) require input of a highly specialized nature and allow an author little control over the form (e.g., layout, textual style, media allocation) of the document. The issue of easy annotation (4) tends not to be addressed.

The next section describes two prototype systems for the authoring of *textual* (section 2) and *multimedia* (section 3) documents that form a suitable starting point for working towards the 'ideal' system outlined above. Key features of this 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 author creates the document by interacting with these feedback texts.

[1] This is a shortened and updated version of Van Deemter and Power (Forthcoming).

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 allows a domain expert to create a KB by editing a *feedback text*, generated by the system, which presents both the knowledge already defined and the options for modifying it. Knowledge is added or modified by menu-based choices which directly affect the KB; the result is displayed to the author by means of a feedback text: thus 'What You See Is What You Meant'. WYSIWYM instantiates a trend in dialogue systems towards moving more of the *initiative* from the user to the system, allowing such systems to avoid 'open' (i.e., unconstrained) natural-language input.

We will focus on applications of WYSIWYM to the generation of documents. The present section focuses on *text* generation: the KB created with the help of WYSIWYM is used as input to an NLG program, producing a document for the benefit of an end user. Present applications of WYSIWYM 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. By interacting with the feedback texts, the author can, for example, specify a procedure for performing a task to the KB, e.g. preparing an inhaler for use. 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 cleaning, storing, and removing something, made visible in 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 exists a procedure p whose type is `Remove`.') From the updated knowledge base, the generator produces a feedback text of the form 'Remove ... from ...':

> 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, including a Cover. 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.

Note that WYSIWYM allows 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). WYSIWYM also allows authors to specify the *form* of the text, by building a second, form-related KB which describes the *style and layout* of the document. This form-related KB may state, for example, that the maximum paragraph length is 10 sentences. 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), where the document says *Remove the cover of your inhaler.* 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 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 document that corresponds to s illustrated. 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

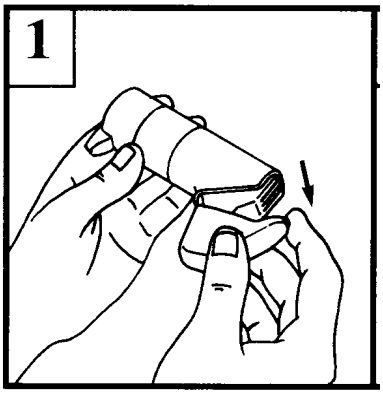

Figure 1: One of the pictures in the library of the authoring system

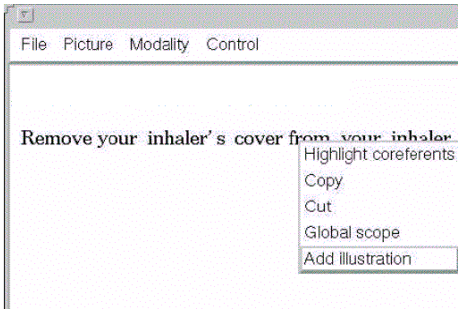

Figure 2: Screendump: Author makes a requests for illustration

inhaler's cover from your inhaler'. In domains where all the pictures are 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 thing 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. Suppose 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 `Actor` is the reader, 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 are 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). Thus, ILLUSTRATE represents the meaning of the picture in Fig. 1, for example, as follows:

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

('There exists a 'Remove' action whose `Source` is a 'haler' (this subsumes various anti-astmatic devices) and whose `Actee` is a cover of the same haler.')

2. How is the library created? This is of great practical importance because the logical complexity of the pictures involved could 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, (virtually) 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. The top of the screendump shows the accompanying feedback text containing anchors for further additions.

3. What is the selection algorithm? Clearly, a picture can illustrate a text without expressing all the information in it. For example, Fig. 1 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.*

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 (Van Deemter 1999).

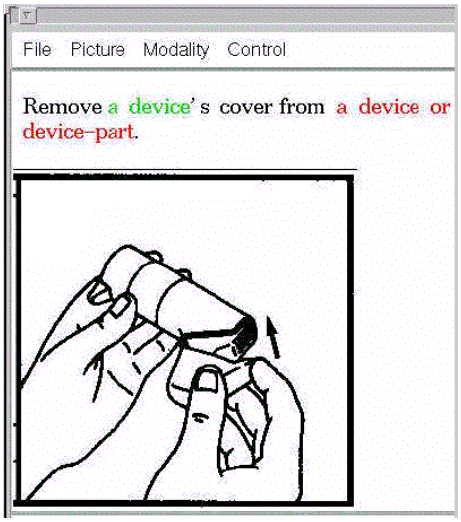

Figure 3: Screendump: A stage during the annotation of a picture

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

- A uniform interface supports editing of all semantic representations, regardless of the medium in which they happen to be expressed.
- When used for the construction of annotations of pictures, the T-Box of the system ensures that only those properties can enter an annotation that are relevant in connection with it. (For example, the height of the patient is regarded as irrelevant, and consequently the T-Box does not make height an attribute of a person.)
- Pictures are retrieved by a simple reasoning process; 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 (cf. Van Rijsbergen 1985, Van Deemter 1999).

4 Conclusion; Future Work Towards the Ideal

The PILLS system (section 2) goes some way towards 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. Among the improvements that we see as most urgent are the following:

- *Media allocation.* 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).
- *Other media.* Little in ILLUSTRATE hinges on the fact that the objects in the library are pictures. The same idea, for example, can be used for annotating sound or *canned text* (for example, a complex fragment of law code, which needs to be rendered *verbatim*). Of great practical interest, finally, is the possibility of including documents authored previously, 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. This could happen, for example, when the text 'remove the capsule from the foil as shown in the picture' (ABPI 1997) is 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.

These extensions hinge on ILLUSTRATE's ability to manipulate semantic representations, where one and the same representation language is used for the different media: a multimedia 'interlingua' (e.g. Barker-Plummer and Greaves 1995). In the case of an author selecting a picture using thumbnails, for example, the semantic representation of the picture enables the author to find a suitable location for the picture and adapt the text by omitting from it information that is now expressed by the picture.

A final extension of the ideas outlined in this paper would involve completing the symmetry between feedback and output: all present WYSIWYM systems use purely textual feedback:

- The content of the KB is rendered in natural language (using sentences like 'Remove **a device**'s cover from your inhaler'), and
- The author modifies the KB by choosing between alternatives that are presented by means of natural language (e.g., by clicking on the word *inhaler*, causing **a device** to be an inhaler).

In principle, however, feedback can be as multimodal as the target document. We are currently exploring the possibility of allowing an author to express some of her choices (more specifically, the choice for a particular referent) by clicking on a mouse-sensitive part of a picture; the system could generate an updated feedback text (possibly along with an updated picture) as a result. In some technologically complex domains, for example, where a brief description of an object which is suitable to appear in a menu may be difficult to obtain, this might lead to a further improvement of the WYSIWYM technique.

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. Greaves. 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.

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, 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).

K. van Deemter (1999). Document Generation and Picture Retrieval. In Proc. of Third Int. Conf. on Visual Information Systems, Amsterdam, Springer Lecture Notes.

K. van Deemter and R. Power. (Forthcoming) Authoring Multimedia Documents using WYSIWYM Editing. To appear in Procs. of COLING-2000 conference.

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.