



ELSEVIER

Computer Standards & Interfaces 18 (1997) 583–593

COMPUTER STANDARDS
& INTERFACES

AutoBrief: a multimedia presentation system for assisting data analysis

Stephan Kerpedjiev^{a,*}, Giuseppe Carenini^{b,1}, Steven F. Roth^{a,2},
Johanna D. Moore^{b,3}

^a Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

^b Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260, USA

Abstract

We present an approach to generating multimedia presentations that integrates hierarchical planning to achieve communicative goals, and task-based graphic design. A planning process decomposes domain-specific goals to domain-independent goals, which in turn are realized by media-specific techniques such as task-based graphic design. We apply our approach to developing AutoBrief, a system that summarizes large data sets using natural language and information graphics. Finally, we analyze AutoBrief in terms of the standard reference model (SRM). © 1997 Elsevier Science B.V.

Keywords: Multimedia presentation; Information seeking tasks; Media allocation; Information graphics; Presentation planning

1. Introduction

The practical problem that we want to solve is assisting analysts and other types of specialists to understand patterns and changes in large data sets and to convey this information to others (e.g., brief their upper management or convey to peers their observations, hypotheses, and conclusions). For example, transportation schedulers often want to know how adding a certain amount of resources affects lateness. To this end, they produce a number of simulations, analyze them, summarize the results, and prepare a concise description of their findings for subsequent use. This description could take the

form of a textual summary of the most important aspects of the data, one or more graphics elucidating an important aspect, or a multimedia presentation combining text and graphics. Performing such kinds of tasks would be greatly facilitated if a tool could automatically extract the relevant pieces of information and present them in an appropriate form.

Our effort to build such a tool, which we call AutoBrief, continues a series of similar projects by other researchers aimed at conceptualizing the design principles of multimedia presentations in a domain-independent way. Among the applications previously addressed are instructions for operating physical devices [1,2], explanations of quantitative models [3], route directions [4], statistical reports [5], and weather reports [19]. The genre we are interested in is explanation of exploratory data analysis, which includes summarizations, comparisons and correlations of data.

* Corresponding author. E-mail: kerpedjiev@cs.cmu.edu

¹ E-mail: carenini@cs.pitt.edu

² E-mail: roth@cs.cmu.edu

³ E-mail: jmoore@cs.pitt.edu

In prior work, two complementary views to automatic presentation generation have emerged. Researchers from the natural language processing community [6,2] focus on the communicative intent of a presentation and model utterance generation as a process of hierarchical planning to achieve communicative goals. In contrast, researchers in graphics view the presentations as interfaces for users to perform tasks, which requires modelling the perceptual and logical operations the user needs to perform [7,8], and building systems that design presentations supporting specific tasks. Designing effective multimedia presentations requires that both types of knowledge be used in the presentation design process, and our work seeks to integrate the planning and task views in a single coherent framework.

In this paper, we first illustrate our approach with a sample scenario from the domain of transportation scheduling. Then we describe the communicative

model and clarify its connections with both the planning process and the graphical tasks. We then work through an example of AutoBrief designing a sample presentation. Next, we outline the graphics generator. Finally, we relate our approach to the standard reference model (SRM) for intelligent multimedia presentation systems [9].

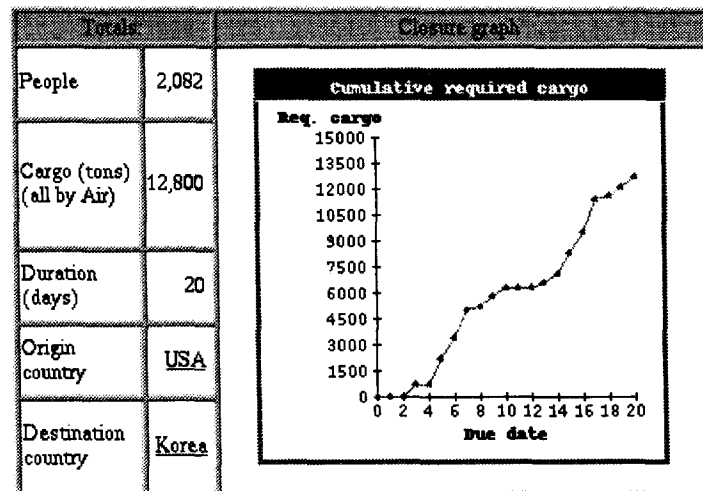
2. Our approach

In our approach, we emphasize four aspects of the explanation of large data sets.

Content planning. The system must select a limited amount of relevant information out of the potentially very large number of facts available in the KB.

Communicative goals direct the system in presenting the content in a way that emphasizes specific aspects, e.g., identifying a particular object or contrasting two facts.

Requirements



Capabilities

- 27 planes with total capacity of 930 tons per day available from day 0 to day 20 ([details](#)).
- Three origin ports, Campbell, Lawson and Tinker, with capacity of 250 tons/day each ([details](#)).
- Two destination ports, Kimpo and Osan, with capacity of 250 tons/day each ([details](#)).

Shortfalls

- Insufficient lift capacity in the date periods 3–5 and 13–15 ([details](#)).
- Insufficient port capacity in Campbell, Tinker and Osan in the date periods 3–8 and 12–18 ([details](#)).

Fig. 1. A summary presentation of a schedule.

Perceptual tasks. Some of the communicative goals can be better satisfied by enabling users to perform certain perceptual tasks on a graphic, instead of simply informing them of the outcome of some automatically performed analysis.

Planning exploratory links. Since our system is intended to support users in performing their analyses, it should enable them to easily request presentations of related information.

The following scenario, which we crafted in HTML and Java, illustrates these aspects in the domain of transportation scheduling. Since during the course of a single day analysts may produce numerous schedules, the first thing they typically want to know about a schedule is summary information about its requirements, capabilities, and possible shortfalls (Fig. 1). This particular selection and organization of attributes is accomplished by a domain-

specific strategy of achieving the goal *know-schedule*.

While most of the attributes in Fig. 1 are conveyed through simple summary statements (e.g., the total number of people), the communicative goals for the attribute *cumulative-required-cargo* are more complex. The user must be able to identify periods of rapid increase in the amount of required cargo as well as dates by which a certain portion of the cargo is scheduled to arrive at the destination ports. Some of these goals cannot be expressed in language as effectively as by the graphic in Fig. 1. The line graph not only enables the user to lookup the values of the attribute (a table could do this as well or even better), but also to scan the development of the graph for steep line segments indicative of rapid increase of the cumulative cargo or flat segments indicative of slow or no increase. The user can also easily

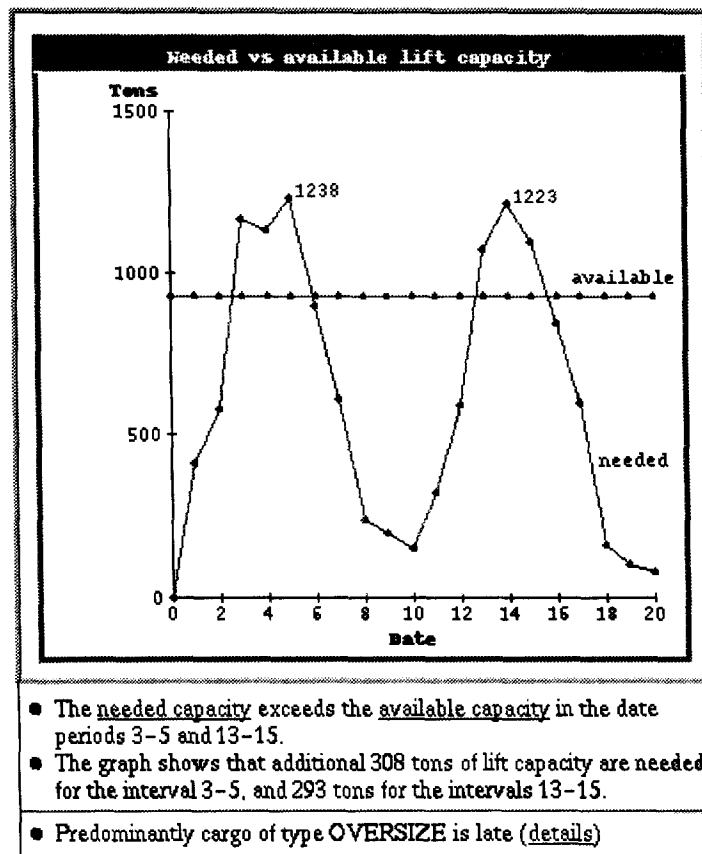


Fig. 2. Comparison of two attributes.

divide the y -axis by a certain portion of the total cargo, find the point where the imaginary horizontal line corresponding to this amount crosses the line graph, and check the x -position of that point, thus, finding the date by which this amount of cargo should be at the destination ports. This presentation illustrates how different communicative goals can be assigned to an attribute and satisfied by enabling perceptual tasks such as *search*, *scan* and *lookup*.

In addition to providing information about various attributes of the schedule, the presentation in Fig. 1 lets the user request more information by making certain portions mouse sensitive (mouse sensitive phrases are underlined in all figures). Associated with each sensitive object, which can be a phrase or a graphical symbol, is a new goal. A mouse click on such an object is interpreted as a request by the user for a presentation that satisfies the goal associated with it. For example, the word ‘details’ right after the sentence saying that the schedule has insufficient lift capacity in two periods (the first bullet in the shortfalls section) is associated with the domain-specific goal of knowing the characteristics of the lift shortfalls. If the user clicks on this word, the system will plan the presentation shown in Fig. 2, which helps the user diagnose the shortfalls.

Planning these hypertext-like links is an important element of our approach that allows the user, after detecting an interesting piece of information, to select a new relevant goal and pose it as a request to the system for a new presentation.

The new presentation (Fig. 2) satisfies the goal *know-lift-shortfall* (the strategy for this goal is explained in detail in Section 4). The two-line graphs allow the user to compare the amount of cargo that the fleet can carry on each date with the expected amount of cargo that needs to be transported on this date. The text makes specific points about the shortfall. For example, the second bullet helps the user answer the questions ‘How much additional capacity is needed and when it is needed?’ The third bullet summarizes the distribution of the late cargo by the observation that predominantly cargo of type ‘oversize’ is late, and enables the user to drill down by clicking on the ‘details’ phrase. As a result, a breakdown of the lateness by cargo type and date is presented graphically (as in Fig. 3) to confirm that the major lateness occurs for cargo of type ‘oversize’

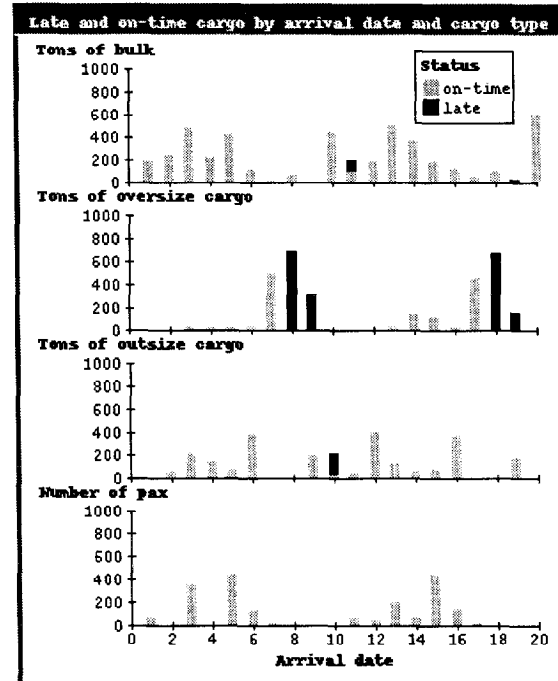


Fig. 3. Correlation among tons of late and on-time cargo, cargo type, and date.

immediately after the periods of lift shortage, and to suggest that insufficient fleet capable of carrying oversize cargo might be the problem.

Thus, with a sequence of three displays, the system helped the analyst to obtain an overview of the schedule, to drill down into lift related information, and to explore a sufficiently refined hypothesis for the cause of the lateness.

3. Modelling the intent of presentations

Planning a multimedia presentation is accomplished by decomposing goals and selecting actions that achieve them. The structure of the goal and action space, part of which is shown in Fig. 4, stratifies into three layers: domain-specific presentation strategies that achieve domain-specific communicative goals; abstract actions that achieve domain- and media-independent communicative goals; and primitive actions that specialize the abstract actions into specific directives to the media generators.

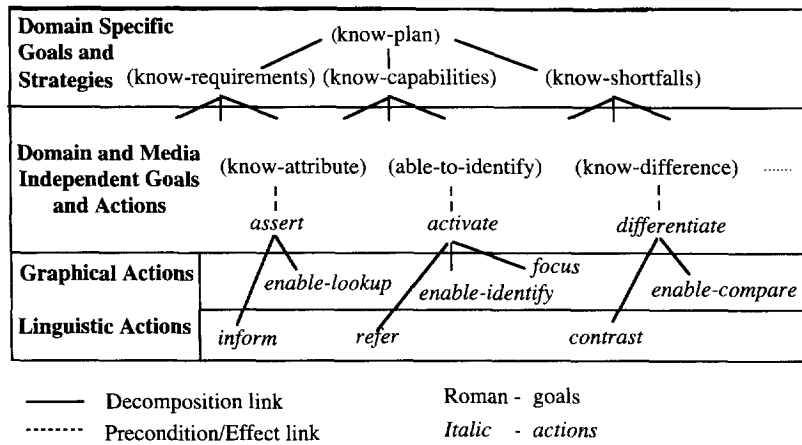


Fig. 4. Goals and actions.

Domain-specific goals represent the desired effects on users produced by a presentation. These goals are decomposed into domain-independent goals by means of domain-specific strategies negotiated with domain experts. Such strategies define the content of the presentation in terms of concepts, relations between them, and the communicative intent associated with them. An example of a domain-specific goal is *know-lift-shortfall* and one of its strategies prescribes that: *know-lift-shortfall* - to know a lift shortfall, the user must know the attributes *daily needed lift capacity* and *daily available lift capacity*, as well as the difference between them; must be able to identify the intervals when the *needed capacity* exceeds the *available capacity*; must know the *maximum needed capacity* in each of these periods, and the *additional cargo* needed to eliminate the shortfall in each of these periods; must be able to request information about the correlation among *late* and *on-time cargo*, *cargo type*, and *date*.

The important elements in this strategy description are the names of the attributes and keywords such as 'know,' 'difference,' 'identify,' and 'request,' which convey the communicative intent of the presentation. Formally, these strategies translate the domain-specific goal into goals at the next level of the communicative model.

Domain and media-independent goals are communicative goals that are common in the genre of exploratory data analysis. Each of these goals is

achieved by a media-independent abstract action. Some goals and the actions that satisfy them are as follows: *know-attribute* - the user knows the values of an attribute for a set of objects, satisfied by *assert*; *know-difference* - the user knows the differences between two attributes, satisfied by *differentiate*; *know-correlation* - the user knows about the correlation of two or more attributes, satisfied by *correlate*; *able-to-identify* - the user can identify each or one of the elements of a set, satisfied by *activate* (The mechanisms for identifying objects in multimedia presentations were studied in detail in Ref. [10].); and *able-to-request* - the user can pose another goal to the system, satisfied by *enable-request*.

The action *enable-request* requires *activate* actions for the objects of the goal, which in turn can be realized in language or graphics. For example, to enable the user to request more information about a particular schedule, the user should be able to identify that schedule (e.g., through a referring expression) and should be given a method for requesting the information (e.g., a mouse click on a mouse sensitive phrase associated with that referring expression).

Linguistic and graphical actions realize the media-independent actions using techniques from the corresponding medium. In text, *assert* is usually realized by *inform*, *differentiate* by *contrast*, and *activate* by building a referring expression (for brevity, *refer*).

In graphics, goals are realized in two ways: by enabling the user to perform certain information-seeking tasks, or by focusing the user's attention on a part of the graphic.

Asserting facts in graphics is realized by enabling the user to perceptually lookup or compute the values of an attribute. (As described later, each task can be supported by various graphical techniques, which are selected by the graphics realization system.) The corresponding system actions are *enable-lookup* and *enable-compute*. In general, lookup is a more efficient task than compute. Depending on the specific graphical technique selected to support the corresponding task, the goal *know-attribute* can be achieved with different levels of accuracy [8]. For example, labels ensure very accurate lookup, while saturation is fairly inaccurate.

Activating objects can be realized graphically in two different ways. If each element of a set needs to be identified, then an attribute that uniquely identifies the individual elements is chosen and encoded by a graphical parameter (e.g., the proper name attribute for a set of people). The action corresponding to this method is *enable-identify*. If a subset must be identified as a whole, then its manifestation on the graphic must be highlighted in a certain way (e.g., using a color or a pointer). The corresponding action is *focus* (e.g., in Fig. 2, the two maxima of the needed capacity are distinguished from the rest by the labels attached to them).

Differentiating attributes is realized graphically by selecting a common encoding technique for those attributes. The corresponding action is *enable-compare*.

Similar techniques based on different information-seeking tasks exist for the other media-independent goals.

4. A detailed example

In this section, we illustrate our approach by describing the design process that results in the presentation in Fig. 2. This presentation fulfills the domain-specific goal *know-lift-shortfall*. A domain-specific strategy decomposes it into the following domain-independent communicative goals: *know-attribute* for needed-lift-capacity; *know-attribute* for

available-lift-capacity; *know-difference* between needed-lift-capacity and available-lift-capacity; *able-to-identify* the intervals of the lift shortfall (where the needed capacity exceeds the available capacity); for each interval of the shortfall, *know-attribute* for the maximum needed lift capacity; for each interval of the shortfall, *know-attribute* for the maximum additional lift capacity necessary to eliminate the lift shortfall; *able-to-request* for the goal *know-correlation* of tons of late and on-time cargo, cargo type and date.

The actions that can achieve these goals are *assert*, *differentiate*, *activate*, and *enable-request*. The next level of decomposition realizes these actions through media-specific ones. We will discuss the way these media independent actions are realized in Fig. 2 and also point to alternative methods of achieving the same goals.

The two *assert* and the *differentiate* actions are realized graphically in Fig. 2 by two *enable-lookup* and one *enable-compare* primitive actions. Since the two attributes *needed-lift-capacity* and *available-lift-capacity* are time series, they were visualized as two line graphs. The common encoding technique for the two attributes is *y-position*. Alternatively, in language, the *assert* for available-lift-capacity could be realized by the sentence 'The daily available capacity is 930 tons,' but the realization of the *assert* for needed-lift-capacity would be awkward, resulting in the enumeration of 20 values. The differentiation is realized linguistically by the sentence in the first bullet in Fig. 2 'The needed capacity exceeds the available capacity in the date periods 3–5 and 13–15.'

The identification of the two intervals is accomplished linguistically in Fig. 2 by the two referring expressions 'periods 3–5 and 13–15'. Alternatively, it might have been realized graphically (action *focus*) by pointers to the two intervals as shown in the bottom part of Fig. 5.

The maximum needed capacity in the two shortfall intervals is asserted in Fig. 2 through the *y-positions* of the two high points on the line representing needed capacity. However, since high accuracy is needed, two labels were added to these points representing the maximum values, 1238 and 1223. Alternatively, the same *assert* actions could be realized linguistically by two *inform* actions.

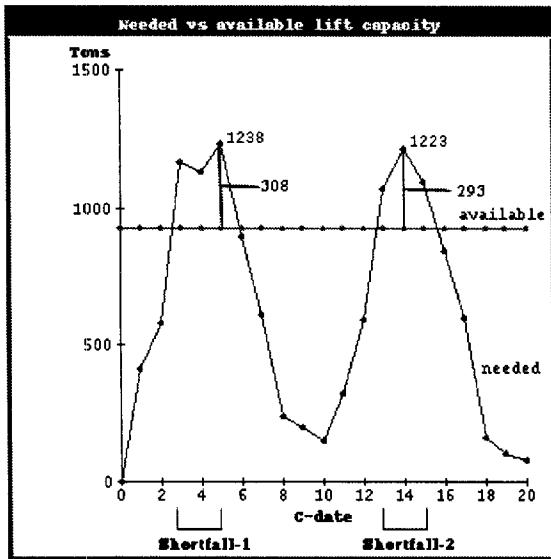


Fig. 5. Asserting the needed additional cargo by graphical lookup and identifying the shortfall periods by focusing.

The additional capacity needed to eliminate the shortfall is not directly encoded on the graphic in Fig. 2. However, it can be evaluated by perceptually computing the difference between pairs of points on the two line graphs. Since this is a very inaccurate way to achieve the *know-attribute* goals, they were realized linguistically in Fig. 2 by the sentence 'Additional 308 tons for the interval 3–5 and 293 tons for the interval 13–15 are needed to eliminate the shortfall.' A possible way to achieve these goals by accurate *enable-lookup* actions is shown in Fig. 5. Two vertical line symbols have been added that identify the maximum differences between needed daily and available daily capacity for the two intervals, and labels have been added for accurate lookup of the values.

Finally, the *enable-request* action for the complex correlation has been realized through the summary statement in the third bullet (Fig. 2) and by appending the mouse-sensitive phrase 'details' to the end.

5. Graphics generation

For graphics design and realization, we use SAGE. It incorporates design rules that apply encoding and composition techniques based on characteristics of

the information to be presented [8] and provides flexible tools for interactive design [12]. For the purpose of this project, we have developed a new tool in SAGE that designs graphics based on tasks that the users should be able to perform. This tool implemented in FUF [13] (the same formalism in which we are implementing the NL generator) performs a grammar-driven search of encoding and composition techniques. The SAGE designer works bottom-up by picking up a task-enabling action, selecting an appropriate encoding technique, and composing it with the techniques for other actions according to the basic design rules of SAGE. In addition to generating the graphic, SAGE returns additional effects that this particular design achieves as well as any complexity metric of the interpretation of the graphic. The former is used for media coordination and follow-up questions while the latter spawns caption generation [14]. SAGE realizes a graphic design through its rendering component, which makes a number of graphical resource allocation and layout decisions.

6. Relation of AutoBrief to the SRM

In this section, we analyze the architecture of AutoBrief in terms of the SRM for intelligent multimedia presentation systems [9]. The SRM abstracts commonalities in the architectures of a number of intelligent multimedia presentation systems (IMMPS) and organizes them in terms of layers, components, and experts (knowledge servers). Furthermore, the SRM assumes that an IMMPS interacts with three external entities: a User, an Application, and a Goal Formulation module. We begin our analysis of the AutoBrief system with a brief discussion of the external entities, and then consider the SRM's layers in more detail. We conclude with a discussion of the expert modules.

User. The user of the AutoBrief system is a data analyst, for example, a transportation scheduler.

Application. The application is a data analysis system, which looks for patterns, relations and trends in the data that may be of interest to the data analyst.

Goal formulation. AutoBrief generates presentations in response to domain-specific communicative goals such as *know-shortfalls*. These goals can be

formulated directly by the application or, as we described in Section 2, by the user. The user provides communicative goals to the system by selecting mouse sensitive objects from the current presentation.

Control layer. The control layer as described in the SRM fulfills two distinct roles: the translation of an input message into a goal understandable by the system, and the ordering in which presentation (sub)goals are processed. In AutoBrief, the first function is straightforward as the output of goal formulation is in a form understandable by the system.

The second function is fulfilled by control strategies that are specified to the Longbow presentation planner [15]. Longbow is a plan-space planner [16], i.e., each node in the plan search space is a partial plan. At each iteration of the planning process, the planner selects a partial plan to refine and a ‘flaw’ (e.g., an unexpanded abstract action, an unsatisfied precondition, or a precondition whose satisfaction has been threatened by ordering of other actions) in this plan to resolve. The process for choosing the partial plan and flaw to work on next is embodied in control strategies that evaluate partial plans and prioritize them in terms of number of steps needed to achieve the goal, complexity of the resulting presentation, consistency with prior presentations, etc. Longbow provides a mechanism for specifying control strategies, but these must be defined by the designer of the multimedia presentation system.

Content layer. In AutoBrief, the functions of the content layer are covered by the presentation planner. As we discussed in Section 3, presentation planning takes place at three levels: the level of domain-specific goals and actions, the level of domain- and media-independent goals and actions, and the level of media-specific actions. Although the levels are conceptually distinct, they are all implemented in a uniform planning formalism.

In terms of the SRM, the domain-specific level fulfills the tasks of goal refinement and content selection as well as high-level ordering of the presentation. The operators at the first level encode high-level domain-specific presentation authoring strategies. For each domain-specific goal there is at least one possible strategy prescribing both how the goal can be decomposed into (partially ordered) lower-level goals, and what content should be in-

cluded in the presentation in order to achieve that goal. Goal refinement is specified via goal decomposition in the operators. The content that is needed to use a particular strategy is specified in the constraints of operators, and satisfying the constraints causes content to be selected.

Eventually, the decomposition leads to domain-independent goals. Operators at the second level prescribe the possible realizations of domain-independent goals in terms of domain- and media-independent abstract actions such as *assert* and *activate*. Finally, the media specific operators decompose media-independent actions into text- and graphic-specific actions. That is, in terms of the SRM, these operators embody media allocation rules.

Design layer. The graphic designer of AutoBrief is SAGE. SAGE parses the final plan looking for graphic-specific actions and designs one or more graphics that support the task-enabling actions. The graphic design in SAGE is a bottom-up process, in which graphic constraints imposed by the task-enabling actions drive the composition of graphical techniques and objects.

The text designer in the first version of AutoBrief performs simple forms of aggregation and lexical choice. The second version of AutoBrief, which is under development, includes an elaborate text microplanner featuring lexical choice, generation of multimodal referring expressions, and theme-rheme organization.

The current version of AutoBrief does not perform any sophisticated layout design. In the final presentation plan, media-specific objects (text paragraphs and graphics) are partially ordered. AutoBrief simply displays media-specific objects according to the sequential structure of the presentation plan. This is clearly an area for future work.

While the presentation planner and the media designers can backtrack internally, currently AutoBrief does not support backtracking from the media design components to presentation planner. This is a limitation, which we intend to eliminate in the next version.

Realization layer. Graphics are realized by the rendering component of SAGE. For each individual graphic, SAGE returns a complex object describing the exact appearance of all graphical objects and the spatial relations between them.

Text is realized using the FUF system [13], which reconciles the structure of each sentence with the SURGE grammar and then linearizes it into a sequence of words with appropriate punctuation marks.

Presentation display layer. All graphical and text objects are rendered using Garnet [11], a general purpose user interface development environment. Interactive techniques, which are used for posing follow-up goals, are realized in the form of Garnet behaviors attachable to either graphic or text objects.

For the next version of AutoBrief, we plan to use the presentation infrastructure of Visage, a rich information-centric environment for data exploration [17].

Expert modules. In AutoBrief two expert modules are used across layers: the application and design experts.

The application expert, which is implemented in Loom, contains definitions for the concepts and relations in the application domain. It mainly serves the presentation planner in performing content layer tasks, but it is also used by the media designers. In particular, the graphic designer needs information about data characteristics to compose expressive and effective graphics, while the text designer needs domain knowledge for lexical choice.

The design expert defines the graphical languages, namely how graphical objects express information. In the first version of AutoBrief, the design expert is used exclusively by the graphic designer, but in the next version, we intend to use it also for caption generation as described in [14].

7. Conclusion

We proposed a framework for integrating decompositional planning and task-based graphic design to generating multimedia presentations. We have applied our approach to developing AutoBrief, an IMMPS that summarizes large data sets using natural language and information graphics. In this paper, we analyzed AutoBrief in terms of the SRM. We found the description of AutoBrief in SRM terms to be quite straightforward. There were only two cases in which the AutoBrief architecture seemed to deviate from the SRM, and in both cases, we were able to reconcile the apparent discrepancy. First, the presentation planner encapsulates components from both the control and content layers. However, we did not

have any problem in determining what planner knowledge sources and planning sub-processes correspond to SRM components (see Ref. [18] in this volume for similar observations about the WIP system). Secondly, SAGE, a single AutoBrief module, designs and renders graphics (two SRM tasks). However, this is just a matter of naming, because SAGE does have a designer and a renderer as separate sub-components that can be mapped to SRM components.

In general, we believe the analysis was helpful and we intend to use the SRM to describe future version of the evolving AutoBrief architecture and to compare it with other IMMPS. We also hope that describing AutoBrief in SRM terms will help other researchers to better understand and possibly reuse our work.

As a final consideration, we believe that the SRM is only a first step in creating a shared set of resources for researchers in IMMPS, and more should be done. We found that the most difficult aspects of designing AutoBrief were the development of the space of goals and actions (Section 3), the specification of knowledge for media allocation, and devising the interface between the space of goals and actions and the design components. We hope the IMMPS research community will focus in the near future on the development of SRMs for these knowledge sources.

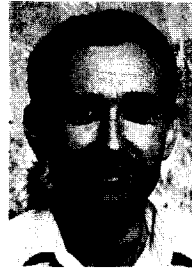
Acknowledgements

This project was supported by DARPA, contract DAA-1593K0005. We are grateful to Mark Derthick and Vibhu Mittal for the numerous discussions and the valuable comments they made on the draft of this paper. In addition, we are grateful for the comments of the anonymous reviewers.

References

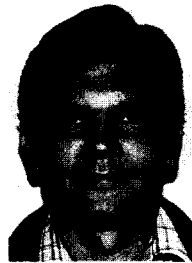
- [1] S. Feiner, K.R. McKeown, Automating the generation of coordinated multimedia explanations, *IEEE Comput.* 24 (10) (1991) 33–40.
- [2] W. Wahlster, E. Andre, W. Finkler, H.-J. Profitlich, T. Rist, *Plan-based integration of natural language and graphics generation*, *Artif. Intell.* 63 (1993) 387–427.

- [3] S.F. Roth, J. Mattis, X. Mesnard, Graphics and natural language generation as components of automatic explanation, in: Sullivan, Tyler (Ed.), *Intelligent User Interfaces*, Addison-Wesley, Reading, MA, 1991, pp. 207-239.
- [4] M.T. Maybury, Planning multimedia explanations using communicative acts, in: *Proc. 9th National Conf. Artif. Intell.*, Anaheim, CA, July, 1991, pp. 61-66.
- [5] M. Fasciano, G. Lapalme, PostGraphe: a System for the generation of statistical graphics and text, in: *Proc. 8th Int. Workshop Natural Language Generation*, Sussex, UK, June, 1996, pp. 51-60.
- [6] J.D. Moore, *Participating in explanatory dialogues*, MIT Press, Cambridge, MA, 1995.
- [7] S.M. Casner, A task-analytic approach to the automated design of information graphic presentations, *ACM Trans. Graphics* 10 (2) (1991) 111-151.
- [8] S.F. Roth, J. Mattis, Data characterization for intelligent graphics presentation, *Proc. ACM Conf. Human Factors Computing (CHI)*, Seattle, WA, April, 1990, pp. 193-200.
- [9] M. Bordegoni, G. Faconti, M.T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, M. Wilson, A standard reference model for intelligent multimedia presentation systems, this volume.
- [10] E. Andre, T. Rist, Referring to world objects with text and pictures, in: *Proc. 15th Int. Conf. Comput. Linguistics (COLING)*, 1994, pp. 530-534.
- [11] B. Myers et al., Garnet: comprehensive support for graphical, highly-interactive user interfaces, *IEEE Comput.* 23 (11) (1990) 71-85.
- [12] S.F. Roth, J. Kolojechick, J. Mattis, J. Goldstein, Interactive graphic design using automatic presentation knowledge, in: *Proc. ACM Conf. on Human Factors Computing (CHI)*, Boston, MA, 1994, pp. 24-28.
- [13] M. Elhadad, Using argumentation to control lexical choice: a functional unification implementation. Ph.D. Dissertation, Columbia University, 1992.
- [14] V.O. Mittal, S.F. Roth, J.D. Moore, J. Mattis, G. Carenini, Generating explanatory captions for information graphics, in: *Proc. 14th Int. Joint Conf. Artif. Intell.*, Montreal, Canada, Aug., 1995, pp. 1276-1283.
- [15] R.M. Young, M.E. Pollack, J.D. Moore, Decomposition and causality in partial order planning, in: *Proc. 2nd Int. Conf. Artif. Intell. Planning Systems*, Chicago, IL, 1994, pp. 188-193.
- [16] D. Weld, An introduction to least commitment planning, *AI Mag.* 15 (4) (1994) 27-61.
- [17] S.F. Roth, P. Lucas, J.A. Senn, C.C. Gombert, M.B. Burks, Ph. J. Stroffolino, J.A. Kolojechick, C. Dunmire, Visage: a user interface environment for exploring information, in: *Proc. IEEE Symp. Info. Visualization (InfoVis '96)*, San Francisco, CA, 1996.
- [18] F. Andre, E. WIP and PPP: a comparison of two multimedia presentation systems in terms of the standard reference model, *Computer Standards and Interfaces* 8 (6,7) (1997) 557-565.
- [19] S. Kerpedjiev, Model-driven assertion-based generation of multimedia weather information, *Bull. Am. Meteorol. Soc.* 76 (10) (1995) 1791-1800.



in information visualization and coordinated text and graphics presentations.

Stephan Kerpedjiev is a project scientist at Carnegie Mellon University where he has worked since 1994. He obtained a M.Sc. in mathematics and a Ph.D. in computer science from the University of Sofia. He held research fellowships at the Institute of Mathematics of the Bulgarian Academy of Sciences, and the NOAA Forecast Systems Laboratory in Boulder, Colorado. He has researched and published in the areas of text analysis and automatic generation of multimedia presentations. His current work is



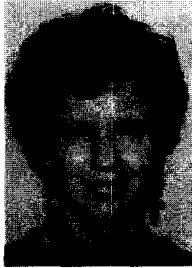
After receiving a Laurea in CS from the University of Milan (Italy), Giuseppe Carenini joined, as a Research Associate, the NLP and Communication Group at IRST (Trento, Italy). At IRST, he developed his interests in natural language generation and interactive systems. He worked there on the design and development of the Alfresco system, an interactive system about Italian frescoes and monuments. In 1992, he joined the Migraine Project at the CS department of the University of Pittsburgh (USA) to study how to design interactive systems following sound knowledge acquisition techniques. Since 1993, he has been a graduate student in the Intelligent System Program at the University of Pittsburgh (USA), working as a graduate research assistant on the Autobrief project. He is also member of the Visualization and Intelligent Interfaces Group at CMU. His current research is on knowledge representation and acquisition for the automatic generation of presentations combining text and information graphics.

After receiving a Laurea in CS from the University of Milan (Italy), Giuseppe Carenini joined, as a Research Associate, the NLP and Communication Group at IRST (Trento, Italy). At IRST, he developed his interests in natural language generation and interactive systems. He worked there on the design and development of the Alfresco system, an interactive system about Italian frescoes and monuments. In 1992, he joined the Migraine Project at the CS department of the University of Pittsburgh (USA) to study how to design interactive systems following sound knowledge acquisition techniques. Since 1993, he has been a graduate student in the Intelligent System Program at the University of Pittsburgh (USA), working as a graduate research assistant on the Autobrief project. He is also member of the Visualization and Intelligent Interfaces Group at CMU. His current research is on knowledge representation and acquisition for the automatic generation of presentations combining text and information graphics.



relational data using coordinated text and graphics. Dr. Roth is applying this visualization technology to a variety of analysis problems with government and industry partners.

Steven Roth directs the Intelligent Visualization and User Interfaces Lab at Carnegie Mellon University. His research is on techniques, environments (e.g., Visage), and automated systems to support information visualization and exploration. The SAGE project has been systems that enable automatic and user-directed design of data visualizations. Related research on AutoBrief is developing systems that automatically generate explanations and summaries of patterns and changes in quantitative and



Johanna Moore is director of the Intelligent Systems Program at the University of Pittsburgh. She received a Ph.D. from UCLA in 1989, and now holds interdisciplinary appointments as an Associate Professor of Computer Science and Linguistics and as a Research Scientist at the Learning Research and Development Center at the University of Pittsburgh. Her research is aimed at developing computational models of the way that humans understand and generate language and graphical representations.

Advances in this research program will improve the communicative capabilities of computer systems that assist, instruct, educate, or represent their users (e.g., expert systems, intelligent tutoring systems, and software agents). Dr. Moore is currently directing federally funded research projects in the areas of natural language interfaces to intelligent tutoring systems, and automatic generation of interactive explanations with coordinated text and graphics. She is the author of *Participating in Explanatory Dialogues* (MIT Press, 1995), and is the recipient of a National Science Foundation Young Investigator Award.