Visualization Techniques for Time-Oriented, Skeletal Plans in Medical Therapy Planning

Robert Kosara and Silvia Miksch

Vienna University of Technology, Intitute of Software Technology
Resselgasse 3/188, A-1040 Vienna, Austria, Europe
{rkosara, silvia}@ifs.tuwien.ac.at

Abstract. In order to utilize elaborate tools and techniques (like verification) for use with clinical protocols, these must be represented in an appropriate way. Protocols are typically represented by means of formal languages (e.g., Asbru), which are very hard to understand for medical experts and lead to many problems in practical use. Therefore, a powerful user interface is needed. We identify the key problems the user-interface designer is faced with, and present a number of "classic" solutions and their shortcomings — which led to our own solution called AsbruView. Its two different views (Topological View and Temporal View) are presented.

1 Introduction and Motivation

Clinical protocols exist for many areas of medical care. Such protocols are typically represented as text, tables, or flow-charts. These representations are far from perfect, however, because they lack a clear concept of time and do not allow automation support for verification or quality assessment. In the Asgaard/Asbru[1] project [12], a number of methods are being developed that deal with problems of clinical therapy planning. The key element of these efforts is Asbru, a powerful language to represent time-oriented, skeletal plans. Asbru has a LISP-like syntax, which makes it unusable for domain experts. Powerful methods are useless, however, when they cannot be used by the people they are intended for. This is why we developed a user interface that gives physicians access to Asbru.

In section 2, we give a short introduction to the key concepts of Asbru. The main challenges in visualizing Asbru, plus some possible solutions are discussed in section 3. Our own approach, called AsbruView, is introduced in section 4. We end up with a conclusion and future plans in section 5.

[1] In Norse mythology, *Asbru* (or *Bifrost*) was the bridge to *Asgaard*, the home of the gods (see also `http://www.ifs.tuwien.ac.at/~silvia/projects/asgaard/`).

2 Asbru Concepts

Asbru is a plan representation language that can capture time-oriented, skeletal plans. In order to understand the specific problems we faced in visualizing Asbru, one must be familiar with some of its basic concepts. These will be described here briefly. For a more detailed description, see [12].

Plan Layout (Actions). The plan body contains plans or actions that are to be performed if the preconditions hold. A plan is composed of other plans which must be performed in sequence, in any order, in parallel, or periodically (as long as a condition holds, a maximum number of times, and with a minimum interval between retries). A plan is decomposed into sub-plans until a non-decomposable plan — called an action — is found. This is called a *semantic stop condition*. All the sub-plans consist of the same components as the plan itself.

Preferences constrain the applicability of a plan (e.g., select-criteria: exact-fit, roughly-fit) and describe the kind of behavior of the plan (e.g., kind of strategy: aggressive or normal).

Intentions are high-level goals that should be reached by a plan, or maintained or avoided during its execution. Intentions are very important not only for selecting the right plan, but also for reviewing treatment plans as part of the ever ongoing process of improving the treatment. This makes intentions one of the key parts of Asbru.

Conditions need to hold in order for a plan to be *started*, *suspended*, *reactivated*, *aborted*, or *completed*. Two different kinds of conditions (called preconditions) exist, that must be true in order for a plan to be started: *filter-preconditions* cannot be achieved (e.g., subject is female), *setup-preconditions* can. After a plan has been started, it can be suspended (interrupted) until either the *restart-condition* is true (whereupon it is continued at the point where it was suspended) or it has to be aborted. If a plan is aborted, it has failed to reach its goals. If a plan completes, it has reached its goals, and the next plan in the sequence is executed.

Effects describe the relationship between plan arguments and measurable parameters by means of mathematical functions. A probability of occurrence is also given.

Time Annotations. Time-oriented planning is centered around Asbru's *time annotations*. A time annotation is defined by seven entities: reference point, earliest starting shift (ESS), latest starting shift (LSS), earliest finishing shift (EFS), latest finishing shift (LFS), minimum duration (MinDu) and maximum duration (MaxDu). Any subset of these parameters may be left undefined. Reference points can be abstract points in time, so each reference point can be considered to be the origin of its own time axis.

3 Visualization Challenges

An enormous amount of work has been done in the field of scientific and information visualization in the last few years, but most of these approaches focus on large amounts of multi-dimensional data. For this kind of problem, a number of good visualizations exist now, that make data accessible [5, 6, 13, 19].

The challenges that the designer of a visualization of time-oriented plans faces are quite different, however. We identified five main problems: hierarchical decomposition of plans, compulsory vs. optional plans, temporal order, cyclical plans, and temporal uncertainty. A more detailed description of these problems plus possible solutions that inspired our approach are given below.

Hierarchical Decomposition. Plans can either be actions ("atomic") or consist of sub-plans. A plan can be reused as a sub-plan of another plan. A successful visualization must be able to communicate this concept.

This part is already satisfactorily solved in our Topological View (section 4.1). As an alternative, a tree view, like it is used for viewing file and directory structures, could be used.

Compulsory vs. Optional Plans. A sub-plan can be used in two different ways: it either *must* be executed (compulsory plan) or it *can* be (optional). While a compulsory plan is easy to understand (and to depict), a way of indicating that a plan is optional is a lot more difficult, especially if it must be different from the representation of temporal uncertainty (see below). A blurred depiction of plans [9] therefore cannot be used.

Temporal Order. In some cases, only the set of plans to be used is known, but not the order in which they will be performed. A way of depicting a plan has to be found where the order in which they are depicted does not necessarily correspond to the order in which they will be executed.

Flow-charts [4, 14] have been proposed for this purpose, but they do not cover parallel plans or sets of plans that can be performed in any order (the latter is possible[2], but only with considerable effort that leads to diagrams that are impossible to read). Additionally, flow-charts scale very poorly, i.e. become unreadable when a large number of plans is defined, and they do not cover the temporal aspect (see below).

[2] By defining one path for every possible permutation of the plans. For n plans, this means $n!$ different paths.

Cyclical Plans. Many actions in medicine are cyclic, for example a treatment every two weeks, or blood tests every morning. It is of little value to display all the many instances of the same action when it is known to be cyclical anyway.

We tried sphere and cylinder metaphors (inspired by [5]), but that did not lead to usable representations.

Temporal Uncertainty. The time a plan takes, but also time spans that are considered for the relevance of symptoms are not defined in terms of exact durations. Therefore, a way of visualizing time spans, where only part of the information (e.g. the minimum duration) is known, must be found. This information may be refined later; this is called a minimum-commitment approach [23].

A related problem is that of temporal granularity. It should be possible to tell to what accuracy a point in time has been defined (e.g. seconds, minutes).

Simple ways of indicating uncertainty can be found in [9, 23], but are very limited. These approaches only tell the reader that the data is uncertain, but not to which degree.

A very versatile, albeit difficult to understand solution to this problem can be found in [20]. While the methodology proposed there is very powerful, it is badly suited for *displaying* more than a few plans, especially when they are to be executed in parallel or when they overlap.

A time annotation in Asbru consists of seven values, and thus can be understood as a point in seven-dimensional space. There are a number of visualization approaches to this kind of problem, the most usable of which are parallel coordinates [6, 7]. They are, however, not useful here since they do not clearly indicate the relations between the different quantities.

The most promising way of visualizing temporal uncertainty are glyphs [15] (or Chernoff faces [2]), which is the solution we finally used.

Further Requirements. Following the “Visual Information Seeking Mantra” [22], an overview should be presented first, so that the user can zoom into the parts he or she wants to examine in greater detail. Details should only be displayed on demand.

Often, one works on a small part of a larger structure, but still wants to know the context this part is in. Three basic ideas are used for this: the Perspective Wall [10], FishEye Views ([3], a similar idea is used in [18]), and stretchable rubber sheets [21]. All of these methods of showing context differ from a simple “lens” in that there is no abrupt break between the magnified area and its surroundings, but a smooth transition. This, in combination with scrolling, makes the concepts very easy to understand and use.

4 AsbruView

AsbruView consists of two very different views, which complement one another. Undefined components are displayed in grey in both views. This is easy to spot because of the heavy use of color (see below).

4.1 Topological View

In the Topological View (Figure 1, [8, 11]), we use a “running tracks” metaphor. Every plan is considered a running track, which the patient runs along while the plan is being performed. When the plan completes successfully, the patient is considered to have passed the finishing line, hence a finishing flag is used to represent the *complete condition* (see section 2).

Although this view has a number of drawbacks (temporal uncertainty is practically impossible to represent), it is, due to its simplicity, very effective in communicating the basic concepts of Asbru. This was tested in a few preliminary scenario-based evaluations [1] we did with our medical experts.

Metaphors from traffic control are used for the other conditions, like a “no entrance with exceptions” sign for the *filter precondition* and a barrier for the *setup precondition*. Since the setup precondition can be fulfilled, the barrier is considered to open in this case. A traffic light stands for the *stop condition* (red light), *suspend condition* (yellow light) and the *reactivate condition* (green light).

Hierarchical Decomposition. Plans can be stacked on top of each other, representing hierarchical decomposition. The sub-plans a plan consists of are put on top of that plan. Each plan has a unique color, which makes plans easier to recognize. It also makes reused plans easier to spot.[3]

Compulsory vs. Optional Plans. Plans that may or may not be performed are displayed with a question-mark texture, while mandatory plans have a plain background. As an alternative, a dotted line can be drawn around optional plans on black-and-white displays (this line can be distinguished from the dotted line which marks the current plan, since, in the latter one, the points move).

Temporal Order. By putting plans next to each other along the time axis, one can indicate that these plans will be performed in this sequence. Parallel plans are aligned along the “parallel plans” dimension (Figure 1). Plans that may be performed in any order are put next to each other more “loosely”, and the containing plan has a groove that the plans can be put into as soon as their sequence is determined.

Cyclical Plans. A circle symbol is used to indicate that a plan is a cyclical plan. The maximum number of repetions can be given, but no temporal aspects, like the minimum or maximum delay between retries.

Temporal Uncertainty is not shown in this view. Because of the perspective distortion, it would be impossible to see the temporal dimension properly.

[3] This is, of course, not true for color-blind people. For this reason, we plan to include an option in our prototype that changes the color-selection scheme so that different plans can be discriminated more easily.

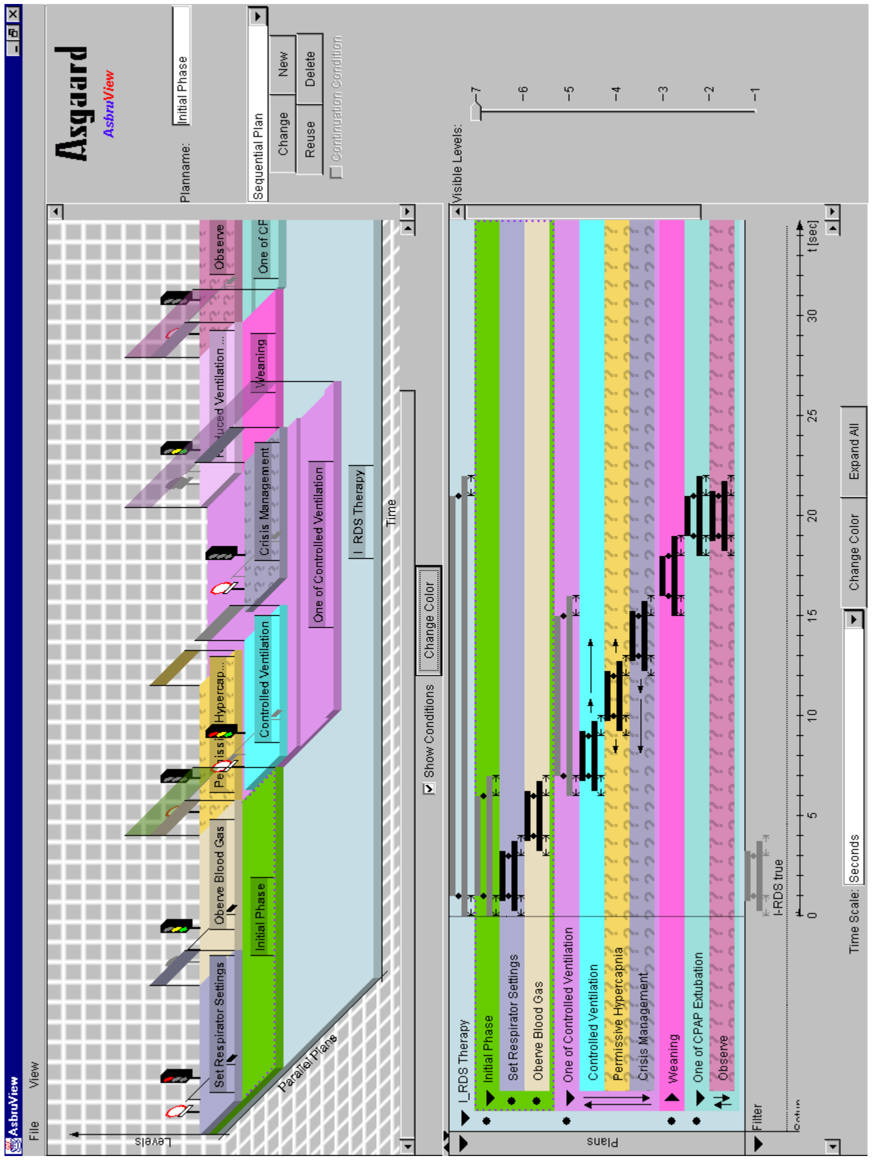

Fig. 1. A screenshot of the AsbruView program. The example depicted is from a real clinical protocol for treating infants' respiratory distress syndrome (I-RDS). The left/upper half shows the Topological View, the right/lower half shows the Temporal View. In the Temporal View, the Plans Facet plus a very small part of the Conditions Facet are visible

4.2 Temporal View

For more complicated tasks, as well as for the experienced user, a more detailed view was developed. It is an extension to the “LifeLines” concept described in [16, 17]. LifeLines are an extension and application of an old concept often named timelines, and presented, for example, in [24].

The idea of LifeLines is very simple: in a diagram with time proceeding from left to right, a horizontal line is drawn for every time span. The lines are drawn in different vertical areas, with a label to the very left of the area. While events whose dates are known (i.e. past events) are captured very well by this approach, it does not deal with temporal uncertainty.

Our own adaptation of LifeLines is described here in a manner similar to section 3 (see Figure 1, lower half).

Hierarchical Decomposition. To make the hierarchical structure of the plans visible, a tree-view-like display is used on the left side. A plan’s sub-plans appear as items underneath one another, bracketed by the containing plan. Similar to the Topological View, each plan has its own, unique color to make identification easier, not only between different facets, but also between the two views.

Compulsory vs. Optional Plans. The same method as in the Topological View is used here.

Temporal Order. A symbol next to every “opened” plan (i.e. a plan whose sub-plans are visible) shows its type. In the example, *I-RDS Therapy* is a sequential plan, *One of Controlled Ventilation* is an any-order plan; a parallel plan would be indicated by two parallel lines.

The order of execution is also indicated by the position of the plans along the time axis. In addition, plans that are to be executed in any order are displayed in one “time slot”, with arrows pointing to other possible execution times.

Cyclical Plans. The first instance of the plan is shown, with arrows pointing to other possible occurrences. If minimum and maximum delays between retries are given, they are displayed in a manner similar to time annotations. The maximum number of retries is given as a number next to the first instance.

Temporal Uncertainty. Instead of simple lines (like in LifeLines), we use an extended version of the time annotation (see Figure 2) we proposed in [11]. The metaphor used here makes the concept of time annotation easy to grasp. All defined components of a time annotation are displayed in black; any undefined components are grey. Additionally, if the LSS or EFS are not defined, the diamonds supporting the MinDu become circles. This means, they can move if the MinDu is changed. It is also easy to understand that the MinDu cannot become shorter than the time span between LSS and EFS, otherwise the “MinDu bar”

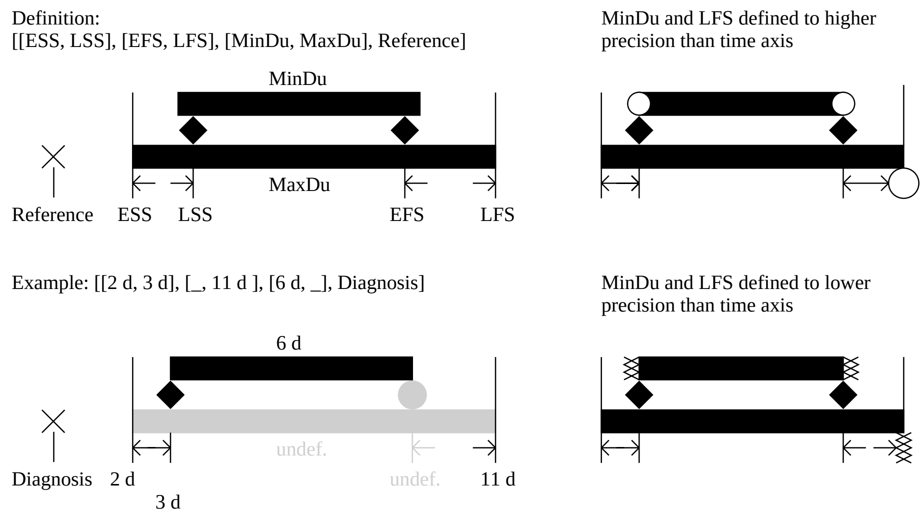

Fig. 2. Time Annotations. On the left side, the definition of time annotations is illustrated (top) and an example given (bottom). On the right side, two cases are depicted in which the time scale of the time annotations is not the same as that of the current time axis

would fall down. Both MinDu and MaxDu are constrained in their maximum length by ESS and LFS: they cannot extend beyond the vertical lines.

Another problem that was not solved for LifeLines is that of different time precision. The user can select the scale of the time axis in a "logarithmic way", i.e. select a granularity of weeks, days, hours, minutes, or seconds. If a point in time is defined to a higher precision than can be displayed with the current time resolution, a circle is put at the corresponding point (Figure 2, top right). If the whole time annotation is smaller than one unit of the current time axis, it is only displayed as one small circle. A similar concept is used in mathematics when one wants to draw a line from A to, but not including, B.

If a point is defined to a lower precision than the current time axis (such as "plus one hour" for a time scale of five minutes), zigzag lines are used to mark the area of "imprecision" (i.e. one half unit of the more precise unit to both sides of the point; Figure 2, bottom right).

Facets. We make heavy use of the "facet" idea [16]. A facet is a vertical region in the display (see Figure 1) that is dedicated to a certain aspect of the data. We are using facets for all of Asbru's aspects: plan layout, preferences, intentions, conditions, and effects. Facets can be opened and closed at any time, and share a common time axis. Thus, the relation between different parts of the display is very easy to understand, and problems from different views showing different parts of a plan, for example, at the same time do not arise. Vertical scrolling of the different facets is independent, however.

Since time annotations play an important role in all aspects of Asbru, the same kind of representation can be used in all facets.

5 Conclusion and Future Plans

We have given a short introduction to Asbru, and presented the main challenges in our efforts to make it accessible to medical experts. A number of possible solutions to these challenges were presented, together with their drawbacks.

The solution to the stated problems, based on many of the listed "possible solutions", was presented. It is called AsbruView, and consists of two views: the Topological View and the Temporal View, offering different ways of interaction.

Most of AsbruView has been implemented, and is currently being evaluated with medical experts.

Acknowledgements

We thank Georg Duftschmid, Klaus Hammermüller, Werner Horn, Christian Popow, and Andreas Seyfang for valuable suggestions and discussions. We also thank Clive Richards and Aaron Sloman for interesting discussions at *Thinking with Diagrams '98*.

This project is supported by "Fonds zur Förderung der wissenschaftlichen Forschung" (Austrian Science Fund), P12797-INF.

References

1. J. M. Caroll. *Scenario-Based Design — Envisioning Work and Technology in System Design*. John Wiley & Sons, New York, 1995.
2. H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68:361–368, 1973.
3. G. W. Furnas. The FISHEYE view: A new look at structured files. Technical Memorandum #81-11221-9, Bell Laboratories, Murray Hill, New Jersey 07974, U.S.A., 12 Oct. 1981.
4. H. H. Goldstine and J. von Neumann. Planning and coding problems for an electronic computing instrument. Part II, vol. 1, U.S. Army Ordinance Department, 1947. reprinted in von Neumann, J., 1963. Collected Works Vol. V. New York: McMillan, pp. 80-151.
5. M. H. Gross, T. C. Sprenger, and J. Finger. Visualizing information on a sphere. In *Proc. Information Visualization*, Oct. 1997.
6. A. Inselberg. Multidimensional detective. In *Proc. Information Visualization*, pages 100–107, Oct. 1997.
7. A. Inselberg and B. Dimsdale. Parallel coordinates for visualizing multidimensional geometry. In T. L. Kunii, editor, *Computer Graphics 1987 (Proceedings of CG International '87)*, pages 25–44. Springer-Verlag, 1987.
8. R. Kosara, S. Miksch, Y. Shahar, and P. Johnson. AsbruView: Capturing Complex, Time-oriented Plans — Beyond Flow-Charts. In *Thinking with Diagrams '98*, University of Wales, Aberystwyth, UK, 22–23 Aug. 1998.

9. A. M. MacEachren. Visualizing uncertain information. *Cartographic Perspective*, (13):10–19, Fall 1992.
10. J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, Information Visualization, pages 173–179, 1991.
11. S. Miksch, R. Kosara, Y. Shahar, and P. Johnson. AsbruView: Visualization of Time-Oriented, Skeletal Plans. In *Proceedings of the 4th International Conference on Artificial Intelligence Planning Systems 1998 (AIPS-98)*, Menlo Park, CA. Carnegie Mellon University, AAAI Press.
12. S. Miksch, Y. Shahar, and P. Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In *Proceedings of the 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*. Open University, Milton Keynes, UK, 1997.
13. S. Mukherjea, K. Hirata, and Y. Hara. Visualizing the results of multimedia web search engines. In *Proceedings IEEE Symposium on Information Visualization 1996*, pages 64–65. IEEE, 1996.
14. I. Nassi and B. Shneiderman. Flowchart techniques for structured programming. *SIGPLAN Notices*, 8(8):12–26, 1973.
15. A. Pang, C. Wittenbrink, and S. Lodha. Approaches to Uncertainty Visualization. Technical Report UCSC-CRL-96-21, University of California, Santa Cruz, Jack Baskin School of Engineering, Sept. 1996.
16. C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: Visualizing personal histories. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, pages 221–227, 1996.
17. C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. Lifelines: Using visualization to enhance navigation and analysis of patient records. In *Proceedings of the 1998 American Medical Informatic Association Annual Fall Symposium*, pages 76–80, 1998.
18. S. M. Powsner and E. R. Tufte. Graphical summary of patient status. *The Lancet*, 344:386–389, 1994.
19. W. Purgathofer and H. Löffelmann. Selected new trends in scientific visualization. Technical Report TR-186-2-97-17, Vienna University of Technology, Computer Graphics, Visualisation and Animation Group, Sept. 1997.
20. J.-F. Rit. Propagating temporal constraints for scheduling. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 383–388, Los Altos, CA, 1986. Morgan Kaufman Publishers, Inc.
21. M. Sarkar, S. S. Snibbe, O. J. Tversky, and S. P. Reiss. Stretching the rubber sheet: A metophor for visualizing large layouts on small screens. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Visualizing Information, pages 81–91, 1993.
22. B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, Washington. IEEE Computer Society Press.
23. D. A. Stevenson, X. Guan, K. J. MacGallum, and A. H. B. Duffy. Sketching on the back of the computational envelope ... and then posting it? In *Proceedings of the Workshop on Visual Representation, Reasoning and Interaction in Design, Fourth International Conference on Artificial Intelligence in Design 1996 (AID'96)*. Stanford University, USA, June 23 1996.
24. E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, 1983.