Visualization of Qualitative Models

Anders Bouwer and **Bert Bredeweg**
Dept. of Social Science Informatics (SWI)
University of Amsterdam, Roetersstraat 15
1018 WB Amsterdam, The Netherlands
E-mail: {anders, bert}@swi.psy.uva.nl

Abstract

To communicate the results of qualitative simulations, good graphics are necessary as well as textual explanations. Qualitative reasoning (QR) provides a basis for automatic generation of graphics because structural relationships between different kinds of entities and quantities are represented explicitly. This paper takes a look at how several QR concepts can be visualized, focusing on certain kinds of information chunks (description views and processes), as well as certain sets of elements (state transition diagrams, causal model). Some ideas have been implemented, which allow domain-independent graphic generation of state-transition diagrams, causal models, process models, and interactive navigation between them. Further work includes layout optimization, aggregation and abstraction mechanisms, and integration of graphics and text generation.

Introduction

In this paper, we study the use of graphics for communicating to users the knowledge captured by simulation models of system behaviour. Considering different representations of such knowledge, it can be useful to distinguish between analogical and propositional representations (Kulpa 1994). Analogical representations are characterised by a parallel or *direct* correspondence between the structure of the representation and the structure of the represented. This in contrast to propositional representations which are seen as less direct and in a sense are more like *descriptions* of the represented. It is argued, and convincingly illustrated with examples, that analogical representations are often easier to understand and reason with for humans, particularly when a 'total' or an 'all-in-one' view of the problem situation is important for solving it (e.g., visual proof of the Pythagoras Theorem). Computers, on the other hand, by their very nature are better in dealing with propositional representations.

Qualitative models and qualitative reasoning have a role in supporting learners in understanding and reasoning about system behaviour (e.g., Bredeweg & Winkels 1998). It is therefore relevant to investigate the use of graphics in relation to this type of modeling and reasoning. From the discussion above at least two questions follow. First, *what* should be represented, and second, *how* should we represent it? As qualitative reasoning is concerned with systems and their behaviour it may seem that as the represented we should focus on these systems (and their behaviour) and develop analogical representations of them. A good example, in this respect, is the Cyclepad system (Forbus *et al.* 1999) where *pictorial* elements are used to communicate knowledge about the structure of thermodynamic systems (components and connections). The pictorial elements used are graphically very similar to the form of the depicted element in reality (or a prototypical version of it). It goes without saying that good analogical representations of the real system are very helpful, maybe even essential, for tutoring purposes. However, there are also limitations. First, these graphics are usually tailored to a specific domain (e.g., thermodynamics) and cannot be used for other domains (e.g., ecology, or medical systems), whereas some of the underlying simulation knowledge might be reusable across domains (e.g., process descriptions of liquid-flow or evaporation). Second, and maybe more important, is the fact that such analogical representations don't show 'the behavioural model' that we (humans) impose on (physical) systems as a means to understand and explain their behaviour. Particularly, notions such as causality are not explicitly captured by such an analogical representation. This is where qualitative reasoning comes in. A qualitative model of a system, and its behaviour, can be seen as a propositional model that captures the knowledge that humans (educators, experts, etc.) have of such (physical) systems; it is a kind of *description* of how we understand it. And it is this 'interpretation or understanding' of the real systems that needs to be communicated to learners. It is therefore that we focus on the qualitative model as the subject of our visualization efforts and not the (physical) system itself.

Having defined our focus, we now turn to the question of *how* to visualize the information. Research on diagrammatic reasoning (e.g., Glasgow, Narayanan, & Chandrasekaran 1995) clarifies how properties of representations can aid human reasoning, but the results of such studies are often tied to specific problem domains and representations. Furthermore, many of these studies don't specify how such representations can be generated automatically. Work on information visualization addresses these problems, but often the approach taken there is either too general for our purposes, encompassing all types of information and representation (e.g., Arens, Hovy, & Vossers 1993, Zhou & Feiner 1996), or too specifically aimed at visualizing large amounts of numerical data (e.g., Roth *et al.* 1997). In our view, research on automatic visualization could take more advantage of the richness of QR knowledge representations.

The goal of our approach is to develop mechanisms which are capable of visualizing qualitative simulation models in a domain-independent way. Our work is based on the GARP simulation environment (Bredeweg 1992), which uses the following building blocks. First, a representation of simple (physical) entities and their structural relations. Second, a representation of time-varying properties in terms of quantities and quantity spaces. Third, representation of all kinds of dependencies between quantities and values of quantities, such as influences and proportionalities (cf. Forbus 1984). Using these building blocks, initial scenarios can be specified as well as libraries of model fragments. Scenarios usually consist of a structural description of the system and some initial values for certain quantities. Model fragments are rule-like, in terms of having conditions and consequences. The former specifies the structural descriptions and the specific quantity conditions that must hold in order for the model fragment to be applicable. The givens of a model fragment specify the behavioural features that can be derived. An important part of this is the specification of the causal model underlying the behaviour.

The ontology underlying the GARP framework is the starting point for our visualization quest. From this we can derive the set of units that we want to find visualizations for. In the next section, we will describe the kinds of information to be visualized in more detail. The majority of the paper will then illustrate how some of this information can be presented graphically. In the discussion section we give an indication of our progress so far, and discuss future directions. The final section concludes this paper with a short summary of our goals and prospective results.

Qualitative Models

What are the things that should be visualized? The GARP framework supplies us with elements of many kinds: quantities, values, derivatives, dependencies, entities, attributes, states, transitions, and model fragments of different types. These elements can be presented as mere labels on their own, or be combined to form larger units for visualization. We distinguish two kinds of such larger visualization units: (1) *chunks*, which combine a (relatively small) number of elements which somehow belong together, and (2) *sets*, consisting of multiple elements, or chunks, of the same kind. To summarize the difference between a chunk and a set, a chunk shows internal structure and detail, while a set gives a structural overview and allows comparisons.

Chunks: collections of elements of various types belonging together according to some viewpoint other than ontological type. Important here is the notion of connectivity in one of the viewpoint dimensions: space (system boundaries), or time (states). Chunks have a core element, and related elements around it. Meaningful chunks in our opinion are:

  Quantity: together with its entity, quantity space, value, and derivative

  Entity: together with all its attributes (quantities)

  State: together with its quantity values, its entities and the model fragments which apply

  Process: together with its triggering inequality and causal dependencies leading to the resulting 'flow' and linking back to the quantities in the inequality statement.

  View: a model fragment with its conditions and consequences. This includes single *description* views, and *composition* views. Both can include entities, quantities, and dependencies, but only the latter can include other model fragments.

Sets: collections of elements (or chunks) of the same type(s), and their interconnections. The set of a particular type of elements presents an overview of the scope and structure of the model. Typical sets we'd like to present as a whole are:

  Causal Model: all quantities, and the dependencies between them (i.e., influences and proportionalities);

  Mathematical Model: all quantities, with their definitions;

  State values description: all quantity values within a certain state;

  System Structure Model: all entities (system elements) and attribute relations between them;

  Is-a hierarchy: all entities, types and supertypes;

  Model fragment hierarchy: this comes in two kinds: the model fragment is-a hierarchy, which consists of all model fragments and their supertype relationships, and the model fragment applies-to hierarchy, which consists of all model fragments, and their applicability subsumption relations;

Visualization of Qualitative Models

What are the basic elements, or *visual primitives* in the graphic representations we want to generate? For different types of elements, we use nodes of different size and shape, notably circles (states), rectangles (entities and model fragments), and oval shapes (quantities). A practical consideration influencing these choices is that rectangles (rounded or not) can incorporate text more efficiently, whereas circles occupy less space, so more of them can be shown (as long as they only contain a short name, or number). Nevertheless, the choice of mapping between element type and visual primitives still is somewhat arbitrary - the main point here is that different types have different shapes so that different types of figures can be easily recognized. Conventions of visual modelling languages outside the field of qualitative reasoning can potentially help here.

For relations, there are a few different kinds of visualization primitives: normal infix mathematical operators (e.g., $+, -, =, <, >$), lines, arrows (for directed relationships), labeled arrows, and encapsulation. Which one is used depends on issues discussed in the following subsections.

State-Transition Graph

First, consider a simple behaviour graph, containing all states and state transitions for a simulation of a piston-system with a heat source. In figure 1, all states are presented, denoted by a number giving the order in which they were generated by the simulation engine. This particular simulation consists of nine states; more complex simulations can easily result in more than a hundred states. In that case, mechanisms are necessary to select an interesting subset, or aggregated view on the whole state-transition graph.

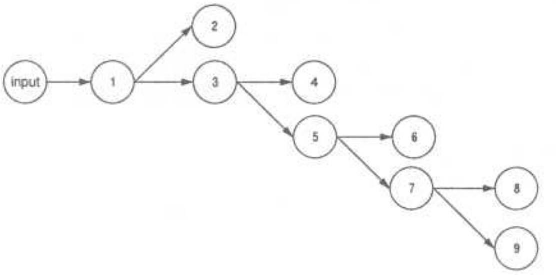


Figure 1: States and state transitions for a simulation of a piston-system with a heat source.

A way to extend this visualization is to include more information in the individual nodes than just the state number. For example, a short list of the most important model fragments present in that state, or a display of the most important quantity value(s). Exactly how to find these 'most important' elements is a question in its own right.

Causal Model

A causal model (sometimes called *dependency graph*, or *influence diagram*) shows all dependencies between the quantities in (a particular state of) the model. The layout of such a graph is important, because it can either hamper or facilitate the recognition of connections, paths, and feedback loops. Besides general principles, like minimal amount of crossing lines, and aligning paths in reading directions as much as possible, structural information from the model can also be taken into account. For example, quantities related to the same entity could be placed close together, perhaps with a bounding box around them. This might facilitate human understanding, by supporting a mapping from the entities in the figure to their corresponding entities in the real world. And in the context of automatic generation, it can greatly reduce the search space for an optimal layout.

When the number of nodes gets large, however, no layout will ever be good enough. Therefore, selection and aggregation are necessary to simplify the picture. The following subsection takes a closer look at part of the causal model, related to a particular process.

Process Visualization

Processes are important in qualitative reasoning because they play a mojor role in explanatory accounts expressing *causality*, the reasons behind change. They offer a focused view on part of the causal model discussed in the previous subsection. The elements selected for a process visualization are the conditions triggering the process (usually an inequality), the process itself, the quantities it directly influences, and the dependencies between them. In figure 2, a heat flow process is visualized, one of the processes taking place in the piston-heater simulation.

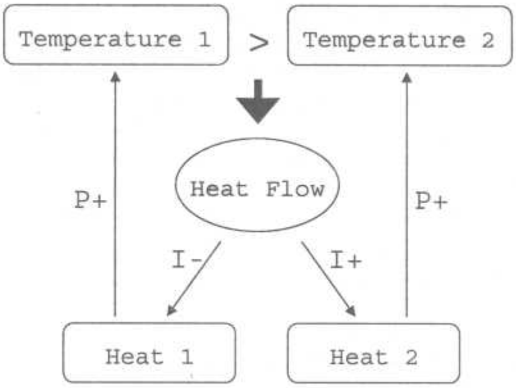


Figure 2: Process model for the heat flow in the first state of the simulation of a piston-system with a heat source.

There are several principles underlying this visualization. The layout of the figure is intended to facilitate a natural reading from left to right and top to bottom. From the top, it starts with the inequality statement which triggers the process (symbolized by a special kind of arrow); in the center, the process is shown with the influences it has on certain quantities, placed below the process; finally, these quantities have proportionality relationships leading back (hence upwards!) to the quantities in the original inequality statement, thereby restoring the equilibrium. Note how this results in a very symmetric figure, with elements of the same kind (e.g., temperature 1 and 2) aligned horizontally, and elements belonging to the same entity (e.g., heat source on the left) vertically aligned. This reflects the fact that the causal paths related to both entities involved in the process have identical, or similar structure.

Considering alternatives for this representation, the structural relationships could even be visualized more explicitly, e.g., by encapsulating the left quantities in a box labeled *heat source*, and the right in a box labeled *gas*. Other elements present in the model but left out of the picture are the specific values of quantities, the definition of the flow rate ($= temperature1 - temperature2$), and the notion of a heat path which is connected to both heat exchanging objects. There is a trade-off between sparsity and ease of interpretation on the one hand and richness of representation on the other. Perhaps the colour dimension could be useful to add more information without cluttering the figure.

A difficulty with this way of visualization is that some processes have more complicated conditions than just an inequality, and don't lead to nice symmetrical figures. It may be that different kinds of processes need slightly different types of visualizations. It is a challenge to invent visualization mechanisms able to capture the essential characteristics of any process, with just enough standardization to facilitate transfer of understanding from one process to another.

Description and Composition Views

Like a process model, a description view is a coherent chunk of information with a conditional part, and information to be added if the conditions hold. Unlike a process, which mainly deals with dynamic aspects, description views describe mostly static aspects. Description views in our piston-heater simulation are *container_view, gas_view, piston_view, world_view* and *heat_source_view*. E.g., the first two make sure that when the simulation scenario includes a container and a heat source, quantities are introduced for the volume of the container, and for the temperature and pressure of the gas, with appropriate values and the appropriate dependencies between them.

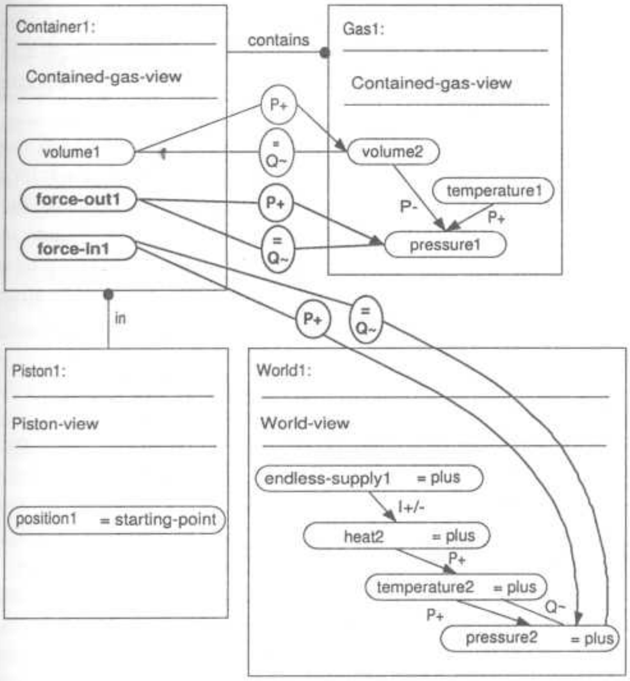


Figure 3: Composition view of the container-piston assembly, including the contained gas view, and description views of the piston and the world.

Composition views are like description views, only more complex, since they describe the aspects of combinations of entities, which are not already covered by the individual model fragments included. An example is the *container_piston_assembly* model fragment, shown in figure 3, which not only includes the models for the *contained_gas* (itself composed of the gas and container views), *piston_view* and *world_view*, but adds information about the force and pressure from inside and outside. To view only the additional information, it should also be possible to hide the contents of each of the included model fragments, and only show clickable references to them.

In the figure, the dependency links are labeled with abbreviation symbols: P+/- for positive/negative proportionality, I+/- for positive/negative influences, Q˜ for quantity correspondence, and = for equal. The conditional part of the model is printed in normal style, while the resulting part is printed in boldface and thicker lines (or in another colour, if possible). In contrast to the process visualization, where the conditional part (the inequality) was placed at the top, the conditional parts in this composition view are too diverse to be placed together at any particular place. Instead, the major structure of the figure is determined by the entities in the model to ensure some intuitive modularity: the container, the gas, the piston and the world. All these entities have quantities, which are displayed within the entity boxes as small oval boxes themselves. This facilitates linking them to other quantities inside the same, or other entities. When their value is known in the composition view, or any of the included description views, it is also displayed inside the oval quantity nodes. The labels for the dependency links are placed next to the link in case the link is between two quantities of the same entity, or placed inside a small circle attached to the middle of the line/arrow in case it links quantities of different entities. The labels = and Q˜ are combined to reduce the number of lines.

The figure is quite complex, but this reflects the complex knowledge structure in the model fragment itself. To enable a user to acquire a mental model resembling this, it will generally be better to show only parts of it first. There are several ways of doing this, e.g., showing each included description view in turn, or showing only entities and structural relationships without quantities and dependencies. As the figures become more complex, textual explanation will become necessary to accompany the visual presentation.

Hierarchy of Model Fragments

The model fragment applies-to hierarchy in figure 4 shows which model fragments include which other model fragments in the piston_heater simulation.

```
container_view(Cont)
gas_view(Gas)
heat_flow((Path, Source, Destination))
heat_source_view(Heat_source)
movement_piston_outwards(Piston)
|  container_piston_assembly((Piston, Cont))
|  |  contained_gas((Cont, Gas))
|  |  piston_view(Piston)
|  |  world_view(World)
movement_piston_still(Piston)
|  container_piston_assembly((Piston, Cont))
|  |  contained_gas((Cont, Gas))
|  |  piston_view(Piston)
|  |  world_view(World)
```

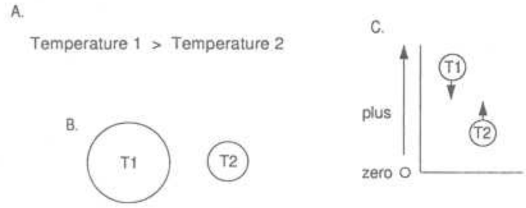
Figure 4: Applies-to Hierarchy of Model Fragments for the simulation of a piston-system with a heat source.

The figure is text-based, with indentation indicating the inclusion relation. Since the goal here is merely to give an overview of all inclusion relations between model fragments, more figurative elements like boxes or arrows would not add much clarity, only cost more space. Vertically oriented trees are also possible, but would generally take up more space, because the width of the tree grows exponentially with respect to the depth of the tree. Similar figures can be generated for the is-a hierarchy of model fragments, and the is-a hierarchy of entities.

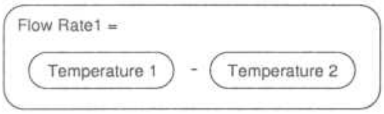
An issue with these kinds of hierarchies is that there may be a lot of repetition involved. For example, the complete structure of `container_piston_assembly` is present in both `movement_piston_outwards` and `movement_piston_still`. A possible solution would be to list all model fragments, together with the model fragments they include, but only at one level deep. This would facilitate lookup of a model fragment to see if it was internally structured, but it would make it harder to see exactly how deep the nesting structure of a composition view is.

Quantities

How quantities should be visualized depends strongly on the communicative goal, and the context in which they are displayed: individual values, quantities in mathematical relations, quantities in state transition graphs, tables, graphs. Several ideas for visualization of quantity values and (in-)equality relations are shown in figure 5.



A, B, C. Three representations of greater(temperature1, temperature2)



D. Definition of the quantity flow rate

Figure 5: Several ideas for visualization of quantities and (in-)equality relationships.

Figure 5A is a textual representation, although it is a different one from the original `greater(temperature1, temperature2)`. Figure 5B is a very graphical representation which may appear very intuitive, but actually brings up some questions: exactly how much bigger should the right bubble be to appear *qualitatively* bigger? If it is clearly bigger, aren't we saying too much? T2 may actually be just slightly greater than T1 in reality. And also, T1 may unjustfully capture more attention than T2, just because it is bigger. Figure 5C is still another version, with both values presented together with their quantity space. In this figure, both variables have attached an arrow indicating the direction of change, which makes it clear that at this moment, T1 is greater than T2 (both being positive), but both are moving towards each other. Figure 5D represents a quantity definition of the flow rate, in terms of a temperature difference. Encapsulating the difference relation within a larger node for flow rate can be useful when the focus needs to be on the flow rate specifically.

Mathematical statements (i.e., equalities or inequalities) can play different roles in the simulation, e.g., a condition for a process or other model fragment, a calculation of a certain quantity based on the values of others, or a constraint on the value of a certain quantity. Because the context of the figure will vary between these cases, the contents will have to be visualized differently as well.

In many cases, single values or relations are not very interesting. Values often need to be compared to values at different places, or different points in time. When the history of a certain quantity is of interest, a table or graph can show how the variable changed over time during a certain state-transition path.

In figure 6, the qualitative values for the position of the piston in the container are shown for a particular trajectory in the simulation. This figure is a graph-like table, with the states in the trajectory along the x-axis, and the values in the quantity space along the y-axis. The value is shown as a symbol indicating the derivative; in state nr. 1 the derivative is zero (0); afterwards it decreases (-).

```
position1(piston1)
[ ] [ ] [ ] [ ] [ ] totally_inside_container
| | | | | | | | | |
| | | | | | | | | | inner_half
| | | | | | | | | |
[0] [-] [ ] [ ] [ ] starting_point
| | | | | | | | | |
| | | | |-| | | | | outer_half
| | | | | | | | | |
[ ] [ ] [ ] [-] [ ] edge_of_container

 1   3   5   7   8  state nr.
```

Figure 6: The qualitative values of the position of the piston for a specific trajectory of the simulation.

A potential hazard with this kind of presentations is that the figure is more specific than the original qualititative propositions. The reader of the figure may erroneously assume that the x-axis represents time, with state successions happening at regular intervals. Another issue is that in the graph, the intervals are reduced to points again, which renders it impossible to judge where in the interval the value lies. If real values are known as well (which is possible in the GARP modelling paradigm), these could be used to properly space the rows and columns to make more accurate judgements possible.

Discussion

Our approach has been partly implemented, in SWI-Prolog /XPCE. Figures of state-transition diagrams (figure 1), entity-attribute diagrams (albeit simpler than figure 3), causal models, entity and model fragment hierarchies (figure 4), quantity definitions (figure 5D) and value tables (figure 6) can already be generated automatically, although some small formatting and layout adjustments have been made by hand.

Our mechanism works for GARP models in different domains, e.g., the piston system with a heat-flow, a balance system with a liquid flow (de Koning 1997), and the ecology of brazilian cerrado populations with different growth and migration processes (Salles, Bredeweg, & Winkels 1997).

Towards Interactive Visualizations

As future work we want to use the static graphics, as described in this paper, and develop interactive visualizations, which allow users to navigate through the large amount of information available, and to investigate things in more detail, or from different viewpoints. Three issues are of importance here: how to navigate, how to select the critical/most interesting aspects of the model, and, how to further enhance the graphics with textual explanation.

The state-transition graph can be a convenient starting point for such interactive visualizations. From there, one can select states, transitions, or paths through the simulation. All states and transitions in the graph have hyperlinks to more detailed information about them: the user can choose between an overview of the entities involved, an overview of the model fragments applying, the causal model, and a table of all parameter values. Depending on the choice, new navigation options become available. All figures also have a link back to their context, so navigation in reverse order is also possible; in that case, the concept looked at in detail is highlighted in the higher level figure.

As already mentioned in several of the above sections, sometimes selection mechanisms are necessary to automatically select the most interesting information elements to visualize. For example, in a state-transition diagram, it can be helpful to show the most important values in every state. For the automatic realization of such support we can build on research dealing with aggregation and abstraction mechanisms (de Koning 1997; Mallory, Porter, & Kuipers 1996).

All figures in this paper contain some textual elements. All of them had a text caption, and were explained in more detail in the text. Further automating this process may benefit from the interesting work that is being done in the area of combining text and graphics, e.g., by Wahlster *et al.* (1993) who focus on planning a complete interactive presentation, and Mittal *et al.* (1998), who are generating figures and their captions automatically.

Conclusion

Assuming that graphics are important in explanations, the goals of this paper have been (1) to show that qualitative simulation models capture a lot of information which can be visualized automatically in a domain-independent way, (2) to argue which kinds of information should be visualized together, and (3) to present some possible graphical representations for such visualization units, with their particular (dis-)advantages.

Some ideas for visualizing the results of qualitative simulation have been implemented already, others have been illustrated by handmade graphics. Further work will include testing the effectiveness of the different visualization options, incorporation of better layout algorithms, mechanisms for abstraction and aggregation, and integration with textual explanation.

References

Arens, Y.; Hovy, E.; and Vossers, M. 1993. The knowledge underlying multimedia presentations. In Maybury, M., ed., *Intelligent Multimedia Interfaces*. MIT Press. 280–306.

Bredeweg, B., and Winkels, R. 1998. Qualitative models in interactive learning environments: an introduction. *Interactive Learning Environments* 5:1–18.

Bredeweg, B. 1992. *Expertise in qualitative prediction of behaviour*. Ph.D. Dissertation, University of Amsterdam, Amsterdam, The Netherlands.

de Koning, K. 1997. *Model-Based Reasoning about Learner Behaviour*. Amsterdam: IOS Press.

Forbus, K. D.; Whalley, P. B.; Everett, J. O.; Ureel, L.; Brokowski, M.; Baher, J.; and Kuehne, S. E. 1999. Cyclepad: An articulate virtual laboratory for engineering thermodynamics. *Artificial Intelligence* 114(1/2):297–347.

Forbus, K. D. 1984. Qualitative process theory. *Artificial Intelligence* 24:85–168.

Glasgow, J.; Narayanan, N. H.; and Chandrasekaran, B., eds. 1995. *Diagrammatic Reasoning: Cognitive and Computational Perspectives*. Menlo Park, California: AAAI Press / MIT Press.

Kulpa, Z. 1994. Diagrammatic representation and reasoning. *Machine GRAPHICS & VISION* 3(1/2):77–103.

Mallory, R. S.; Porter, B. W.; and Kuipers, B. J. 1996. Comprehending complex behavior graphs through abstraction. In Iwasaki, Y., and Farquhar, A., eds., *Proceedings of the Tenth International Workshop on Qualitative Reasoning*, 137–146. Menlo Park, CA, USA: AAAI Press.

Mittal, V.; Moore, J.; Carenini, G.; and Roth, S. 1998. Describing complex charts in natural language: A caption generation system. *Computational Linguistics* 24(3):431–467.

Roth, S. F.; Chuah, M. C.; Kerpedjiev, S.; Kolojejchick, J. A.; and Lucas, P. 1997. Towards an information visualization workspace: Combining multiple means of expression. *Human-Computer Interaction Journal* 12(1/2):131–185.

Salles, P.; Bredeweg, B.; and Winkels, R. 1997. Deriving explanations from qualitative models. In du Boulay, B., and Mizoguchi, R., eds., *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. Japan, Osaka: IOS-Press/Ohmsha. 474–481.

Wahlster, W.; Andre, E.; Profitlich, H.-J.; and Rist, T. 1993. Plan-based integration of natural language and graphics generation. *Artificial Intelligence* 63:387–327.

Zhou, M. X., and Feiner, S. K. 1996. Data characterization for automatically visualizing heterogeneous information. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '96), San Francisco, CA, October 28-29, 1996*, 13–20.