Visualizing Time Dependent Semantics: An Application to Quantum Algorithms

Nivedita R. Kadaba
University of Manitoba
nrkadaba@cs.umanitoba.ca

Pourang P. Irani
University of Manitoba
irani@cs.umanitoba.ca

Michel Toulouse
University of Manitoba
toulouse@cs.umanitoba.ca

Abstract

We have developed a visual syntax for representing concepts that are contingent on temporal properties (time-dependent semantics). A within-group (N=24) experiment was conducted to identify the representations that conveyed best a given semantic. We then applied our representations to the visualization of algorithms in quantum computing and carried out a second experiment (N=16) on subjects unfamiliar with the semantic concepts that were tested. The results show that our representations are intuitive and facilitate a good level of understanding of the algorithms.

Keywords--- Animations, structured-based perception, algorithm visualization, semantic visualization, quantum algorithms.

1. Introduction

A large amount of the information used in the information sciences consists of abstract concepts. In areas such as software engineering or algorithm design, the meaning or semantics of the concepts have to be properly understood in order to construct robust and efficient systems. A significant amount of these semantics are presented visually, through node-link diagrams, such as UML (Unified Modeling Language). These visual representations are used in many cases to facilitate communication between end-users and developers. As a result, the proper development of systems depends on the amount of information that such visual representations are capable of intuitively conveying.

On a strictly temporal dimension, a significant number of semantics can be classified into two general categories. The first category consists of semantics that are independent of time, referred to as *time-independent semantics*. In software engineering the concepts of inheritance, dependency, and aggregation are examples of time-independent semantics. In contrast to time-independent semantics, we can identify a set of semantics that describe the behavior of a system or entity over a period of time, referred to as *time-dependent semantics*.

Time-dependent semantics have been commonly used for describing the dynamic and behavioral aspects of software systems, algorithms, and networks. In this work we focus on creating appropriate representations for a small set of time-dependent semantics that can be used to represent concepts in various areas. We however focus our evaluation on an area that explicitly makes use of these general semantics: quantum computing.

We briefly present the relevant results in perception research and in the area of animated visualizations before describing our results.

2. Related Work

2.1. Visualizing Time-Independent Semantics

According to theories of structure-based perception, the human visual system recognizes objects in our environment by first decomposing them into primitives or blobs [3]. After decomposition, a set of rules that describe the relationships between the primitives are used to perform entry level classification. These rules or relations primarily contribute to object recognition. The relationships between the primitives preserve their two-dimensional silhouette structure, are robust under viewpoint transformation, and are categorical [3].

To a certain degree, semantic information can be expressed in a natural way by mapping it onto the relationships (described above) that can exist between primitives. Irani et al. [6] modeled a set of semantics to show the relationships between the entities in a system. A subset of these semantics, which is relevant to our work, is described here:

- **Generalization or Of-the-Same-Kind** – Primitives with the same structural geometric composition (or shape) can be used to denote objects of the same kind (derived from SIM).
- **Dependency** - If object A is on-top-of object B this suggests that A is supported by B (derived from VER).
- **Multiplicity** - to show multiple associations between two entities, a series of attachments can best denote such a relationship (derived from MUL).

When the perceptual notations are used for constructing diagrams, users are capable of identifying the relationships with very little effort. In their study, Irani et al. [6] compared users' capacity to identify relationships in diagrams containing the perceptual notation to diagrams created with arbitrary notations. Their results show that subjects were 30% more accurate in identifying relationships in the diagrams created with the perceptual notation than in those without.

Proceedings of the Ninth International Conference on Information Visualisation (IV'05)
1550-6037/05 $20.00 © 2005 **IEEE**

The above mentioned perceptual notations are well-suited for representing semantics that are invariant over time. However, many semantics that occur naturally are contingent on temporal properties. If visualizations should correspond to the internal structure or to the content being represented [14], then we hypothesize that some forms of simple animations may be capable of conveying time-dependent semantics. Several studies, which have investigated the efficacy of using animation for representing information, are described in the next section.

2.2. Animation for Representing Information

Intuitively, animations seem to be the most natural way of conveying concepts that change over time. Animated representations have been used in several contexts, particularly as learning aids [1, 2], for showing causal relationships [5, 16], for supporting visual queries in large diagrams [15], or for interacting with hierarchical visualizations [11, 12]. While there are not many applications that use animations for depicting temporal data, the ones that do, show positive results.

Baecker [1], utilized animation to describe the working details of various sorting algorithms. The main motivation behind using animation in [1] was the difficulty in explaining dynamic sequences to students using static diagrams. The results of Baecker's study suggested that although students were able to understand dynamic phenomena, the improvement in understanding was significant when visual simulations were used to enhance the concepts.

Several studies report on the benefits of controlling the animation by means of interactivity. This is necessary if a user is to understand properly the evolution of the semantic over time. A study by Byrne et al. [4] evaluated the effect of animations on learning algorithms. They found that a high level of interactivity, which allowed the learners to control the animation, was more important than animation without interaction. They concluded that interactivity with animation constituted a necessary and integral part of the learning process.

From the large number of studies that have adopted animation as a means for displaying information, a subset of these studies use animations to show time-dependent information. Jones et al. [7] investigated the effect of animated diagrams to show temporal relationships in the flow of blood through the heart. Their results showed that when animations were used rather than static representations, more information was understood by observing the temporal relationships between the working of the heart and the valves.

While animation has enhanced comprehension in certain cases, there is evidence that static representations can be as good as animations. Tversky et al. [13] have shown that static arrows are as effective as animations in showing temporal sequence and direction of motion. In a study by Reiber et al. [10] animated images were shown to be incapable of conveying Newton's laws of motion with significant improvement over the static representations. These studies show that in some cases change over a period of time can be represented by static images. However, the changes in these studies were of a sequential nature.

Most types of animations in previous studies consisted of dynamically changing the spatial location of the objects, of highlighting parts in a temporal sequence or of adding and removing items from the display (these are all defined as animation as some aspect of the display is being dynamically updated). As will be described in the next section, the representations we have constructed consist of smooth transitions that change one or two visual attributes of the object being animated (tweening), to denote a semantic. The spatial location and the direction of motion are not critical for representing the semantics we have selected to visualize. We are particularly interested in determining whether the change of the visual attribute(s) and the visual relationship between objects will lead to the understanding of the semantic being represented. Hence, the animation will be used as a medium for depicting the change of the visual features over some arbitrary time frame.

3. Representing Time-Dependent Semantics

We carried out our investigation in a three-stage process. In the first stage, we constructed different visual representations for the following three specific time dependent-semantics:

- **State Transition** – change in state of an object over a period of time.
- **Interdependence or Entanglement** – interdependency between objects over a period of time.
- **Multiple States or Superposition** – existence of an object in multiple states over a given period of time.

In each case we used a perceptual principle to construct at least one of the instances. The other members of the set were made up of what we thought were reasonable alternatives. In the second stage, we conducted a multipart evaluation study to determine if the subjects agreed with us on our choice of mappings. We conducted the experiment on 24 volunteer students, 12 of whom were familiar with these semantics through a graduate course in quantum computing (experts). The remaining 12 students had not been exposed to these semantics through any formal course (novices). In the third stage, we validated the best mappings by combining them into diagrams that described quantum algorithms. We conducted an experiment on 16 student volunteers, all novices, and evaluated how well the users perceptually understood our representations (described in section 4 below).

For clarity, we have separated each of the sub-experiments for each of the time-dependent semantics, together with the evaluations results, into the following sections.

3.1. State Transitions

In general terms, we infer the semantic of a state transition when an object changes state over time. State transitions occur naturally in our environment, for example when water changes state from liquid to solid or to gas. Often, program objects have internal states, which change as an algorithm is executed.

The semantic of state transition suggests that an object changes its belonging from one class to another class over a period of time. This means that at time t_0 the object belongs to class A and at time t_n the same object belongs to class B. Same shape primitives can be most effectively used to classify objects with a category [3, 6, 8]. From this we hypothesize that state transition can be depicted by a smooth change of object shape.

3.1.1. Representing State Transitions

To test our hypothesis, we created four representations for showing state transitions. Each representation changes one property of the object in a smoothly animated manner (also referred to as tweening). The representations are: Color change (Figure 1.a), Shape change (Figure 1.b), Orientation change (Figure 1.c), and Size change (Figure 1.d).

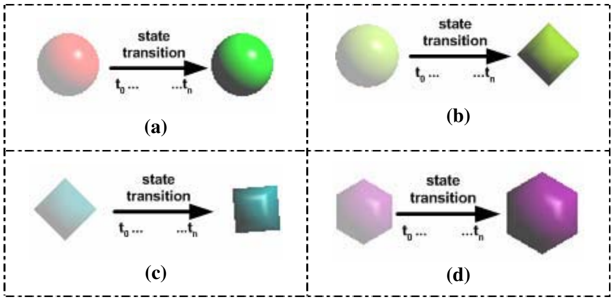

Figure 1: Depicting state transition by changing (a) color, (b) shape, (c) orientation, (d) size. The arrows were not used in the original presentation and are simply included to show the flow of the animation.

3.1.2. Evaluating State Transitions

The experiment was described in a general manner so that a novice (a participant who is unfamiliar with these concepts) would also understand the concepts. The four representations for state transitions were then shown to the participant. The participant was allowed to run each animation several times, as needed. The participant was then asked to rank the representations according to their perception (4 was given to the most favored representation and a rank of 1 was given to the least favored one).

Each set of representations for state transition was shown three times to the participant, on three different screens, with randomized colors and shapes to minimize learning effects.

A top-down test of correlation on the average rankings shows a strong agreement between all 24 subjects for the best-ranked representations (P-value < 0.0001 for null hypothesis of no correlation between rankings chosen by 24 subjects). As seen in the chart above, state transition is best depicted using a "change of shape" representation (B). The results have been depicted in Table 1.

Table 1: Average rankings of state transitions (4 equals best, 1 equals worst)

	Color	Shape	Orientation	Size
Average Rankings	2.87	3.32	1.86	1.97

The results of this part of the experiment support the hypothesis that a *change in shape* can be used to represent a state transition.

3.2. Interdependence

Interdependence is a common semantic that manifests itself over a period of time. This relationship is common and exists, for example, between employers and employees or between variables in a system.

The semantic of interdependence suggests that two objects, over a period of time, become interdependent on one another. Irani et al. [6] suggest that dependency between two objects can be represented by placing them in proximity to one another, typically with the dependent on-top-of the depended. Based on this, we hypothesize that spatial proximity can also be used to represent interdependence between two objects.

3.2.1. Representing Interdependence

We constructed four representations for the semantic of interdependence, as Figure 2 illustrates. To show that two objects become interdependent, the representations consisted of (a) change to a common color, (b) smoothly inserting a connection between two objects, (c) change to a common shape, and (d) moving two objects closer to each other and partially meshing them to each other.

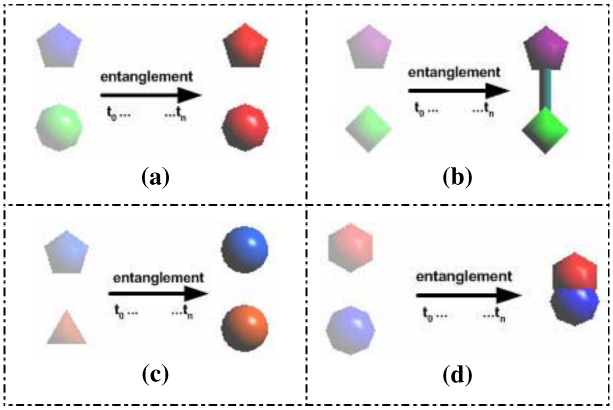

Figure 2: Representing interdependence by (a) change to common color, (b) creating a connection, (c) change to common shape, (d) proximity with partial inter-meshing.

3.2.2. Evaluating Interdependence

The same steps as those used for evaluating state transitions were performed.

A top-down test of correlation on the average rankings shows a strong agreement between all 24 subjects for the best-ranked representations (P-value < 0.0001). As seen in Table 2, interdependence is best depicted using "proximity" (D).

Table 2: Average rankings for interdependence (4 equals best, 1 equals worst).

	Color	Connection	Shape	Proximity
Average Rankings	2.11	2.72	1.96	3.22

The results support our hypothesis in that *proximity with partial inter-meshing* of objects can be used for representing interdependence.

3.3. Multiple States or Superposition

The semantic of superposition occurs when an object exists in multiple states, simultaneously. In its essence, superposition represents the concept of multiplicity that develops over a period of time. This semantic occurs in several contexts. For example, a parent process can spawn multiple child processes. This happens dynamically and creates a one-to-many relationship over the execution of a multi-threaded program.

One-to-many semantics can be depicted using multiple objects [6, 9]. Since superposition is an extension of the concept of multiplicity, we hypothesize that multiple objects can be used for representing this semantic.

3.3.1. Representing Multiple States

We created three representations to depict the relationship of multiple states. To show that an object is in multiple states, we used the representations of (a) multiple duplicates, (b) multiple containments and (c) multiple merged shapes, as shown in Figure 3.

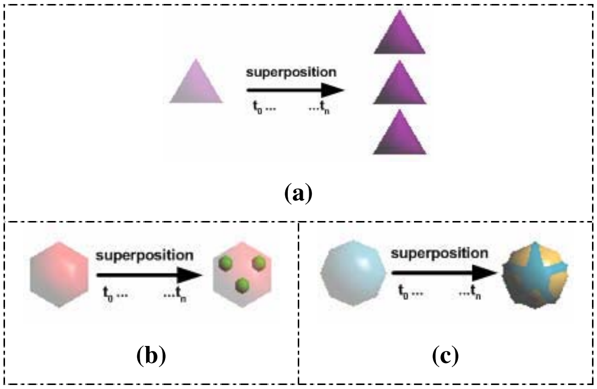

Figure 3: Representing multiple states by (a) multiple duplicates, (b) multiple containments, (c) multiples merged.

3.3.2. Evaluating Multiple States

The same steps as those used for evaluating state transitions and interdependence, were performed.

A top-down test of correlation on the average rankings shows a strong agreement between all 24 subjects for the best-ranked representations (P-value=0.026). As seen in Table 3, multiple states are best depicted using "Multiples" (A).

Table 3: Average rankings for multiple states (3 equals best, 1 equals worst)

	Multiples	Multiple Containments	Shape Merge
Average Rankings	2.61	1.95	1.44

These results confirm our hypothesis that *multiples* can be used for representing multiple states.

The visual vocabulary we created for the time-dependent semantics is linked to a set of perceptual semantics via the representations used for depicting time-independent semantics. This was possible since the meanings for the time-independent semantics strongly resembled the concepts explicated using the time-dependent semantics.

4. Validation

The semantics of state transition, multiple states and interdependence all occur within the framework of quantum algorithms. Hence, to validate the representations described above, we tested their capacity for eliciting information in quantum algorithms. We first briefly outline some elements of quantum algorithms and then describe the results of our experiment.

4.1. Quantum Algorithms

Quantum computing has emerged as an important interdisciplinary field, merging theories in mathematics, physics and computer science. So far, a significant portion of research in quantum computing has focused on the design of quantum algorithms. These algorithms exploit quantum phenomena such as non-locality of quantum systems, superposition of states, quantum interference, and entangled quantum systems, to perform information processing. Quantum phenomena are usually difficult to understand intuitively and often require complex mathematical descriptions. As a consequence, it is a challenge to properly assess the results of the steps throughout the execution of quantum algorithms.

A commonly used representation for quantum algorithms is a circuit, similar to the one found for representing classical logic circuits. In a circuit diagram, nodes represent operations and links connecting the nodes represent the state of the quantum register before and after the execution of a quantum operation. In quantum circuits, nodes are referred to as quantum gates. Qubits in the quantum register are identified using the *Dirac notation* as $|0>$ or $|1>$ when in a basis state and as $a|0> + b|1>$ when in a superposed state (a and b are real coefficients such that $|a|^2 + |b|^2 = 1$). The entire configuration and connections of quantum gates represents a given algorithm.

To properly understand the inner workings of quantum algorithms, several key concepts or semantics need to be identified and understood. A significant level of expertise is required to understand the outputs at each execution stage of an algorithm. Furthermore, existing tools and notations lack the ability to facilitate the understanding of quantum algorithms.

We evaluated our syntax in the context of quantum algorithms. In particular, we were interested in determining whether our visual notations facilitate intuitive identification of a concept. Hence, we conducted an experiment, which is described below.

4.2. Experiment

The first experiment focused on short-listing a set of representations that were simple, but effective enough to describe various time-dependent semantics. The first experiment provided a list of “best” representations for the previously selected semantics.

In order to evaluate the above visual syntax, a second experiment was conducted to evaluate the effectiveness of the mappings in a typical educational scenario. The setup, description and results of the experiment are described below.

4.2.1. Participants

18 students from a local university participated in this evaluation. The factors for this experiment were the display type (perceptual notation vs. quantum notation) and the error rate, which was measured by assessing the effectiveness of the notation in conveying the semantic. None of the subjects had any previous experience with the semantics used in this study.

4.2.2. Materials

Four circuits (or four algorithms) were constructed for the experiment. The circuits differed in complexity, i.e. two of the circuits contained more algorithmic steps than the other two, and notation type, i.e. two of the circuits displayed the output in a quantum notation and the other two displayed the output in a perceptual notation. Each circuit consisted of several gates which acted upon the inputs to the circuit. The experiment used a 2x2 (2 notation types, text and graphics, and 2 levels of complexity, simple and complex) within-subject design. A snapshot of the experiment has been shown in Figure 4.

4.2.3. Procedure

Each of the participants were shown the four circuits in a Latin square fashion and were asked to answer a series of multiple choice questions. There were three concepts that were shown to the students in different combinations; change in state, change to multiple states, and change to combined states. At the completion of the experiment, the participants were asked to fill out a subjective questionnaire, to serve as experimental feedback.

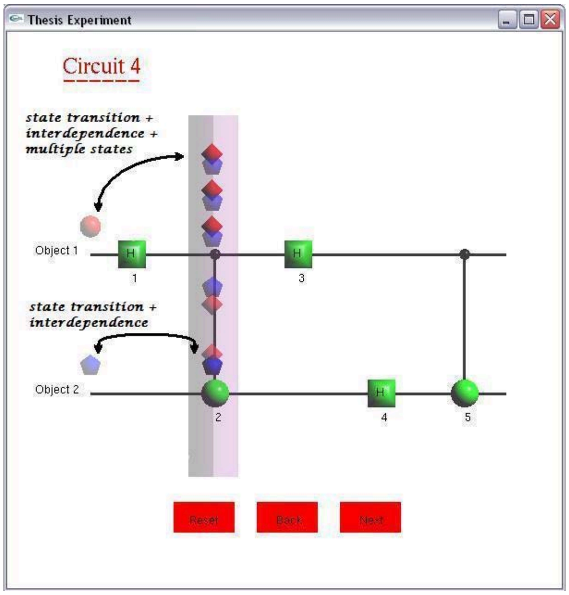
Thesis Experiment
Circuit 4
state transition + interdependence + multiple states
Object 1
H
1
H
3
state transition + interdependence
Object 2
2
H
4
5
Reset
Back
Next

Figure 4: Arbitrary complex algorithm (annotations were not included in the experimental setup).

4.2.4. Results and Discussions

Each subject was given a score of 1 if they matched the correct representation to the semantic or a score of 0 otherwise. Results are summarized in Table 4, which reports error rates by algorithm size. The results are obtained by averaging each subject’s scores. A One-Sample T-Test (or Sign Test) statistically shows that overall subjects performed better with the perceptual notation ($p < 0.01$). Subjects performed slightly better with the quantum representation over the perceptual notation on small circuits. However, the difference is not statistically significant (p-value=0.403). For large circuits, subjects performed significantly better with the perceptual notation than the quantum representation ($p < 0.001$).

Table 4: Error rate in matching the animation to the correct semantic.

	Quantum Notation	Perceptual Notation
Simple	11.46%	18.75%
Complex	61.31%	24.65%

Our results suggest that the perceptual notation is particularly useful for more complex quantum algorithms.

5. Conclusion

This paper reports on the construction and evaluation of a set of visual representations for showing semantics that are dependent on time. Our animations are based on visualizations of time-dependent semantics that represent similar concepts.

The results of our first evaluation show that novices and experts selected the same basic representations for the chosen set of semantics. To further evaluate our representations, we conducted a second experiment to validate the syntax. The syntax was used for representing concepts of quantum algorithms. The results of our evaluation, performed on novices, show that our representations show significant improvement over the quantum representations in a complex scenario.

The visual representations constructed and described in this paper can be used to represent complex concepts or time-dependant semantics. However, in order to make them effective, interaction techniques need to be developed for controlling the visualizations. Further work is being conducted to evaluate the effectiveness of different interaction techniques to assist users of these semantics.

6. Acknowledgements

We thank Colin Ware for reviewing and making suggestions on improving this article. This study is partially funded by an NSERC grant.

7. References

[1] R. Baecker, *Sorting out Sorting : A case study of software visualization for studying computer science.*, in *Software Visualization : Programming as a Multimedia Experience*, John T. Stasko, J.B.D., Marc H. Brown, Blaine A. Price, Editor. 1998, MIT Press. p. 369 -381.

[2] R.S. Baker, M. Boilen, M.T. Goodrich, R. Tamassia, and B.A. Stibel. *Testers and Visualizers for teaching data structures*. in *The proceedings of the thirtieth SIGCSE technical symposium on Computer Science Education*. 1999. New Orleans, Louisiana, United States: ACM Press.

[3] I. Biederman, *Recognition-by-Components : A Theory of Human Image Understanding.* Psychological Review, 1987. **94**(2): p. 115-147.

[4] M.D. Byrne, R. Catrambone, and J.T. Stasko, *Evaluating animations as student aids in learning computer algorithms.* Computers and Education, 1999. **33**(4): p. 253-278.

[5] N. Elmquist and P. Tsigas. *Causality Visualization Using Animated Growing Polygons*. in *2003 IEEE Symposium of Information Visualization*. 2003. Seattle, Washington.

[6] P. Irani, M. Tingley, and C. Ware, *Using Perceptual Syntax to Enhance Semantic Content in Diagrams.* IEEE Computer Graphics and Applications, 2001. **21**(5): p. 76-85.

[7] S. Jones and M. Scaife, *Animated Diagrams : An investigation into the cognitive effects of using animation to illustrate dynamic processes*, in *Theory and Applications of Diagrams. Lecture notes in Artificial Intelligence*, Anderson, M. and Cheng, P., Editors. 2000, Springer-Verlag: Berlin. p. 231-244.

[8] D. Marr, *Vision : A computational investigation into the human representation and processing of visual information.* 1982, San Fransisco, CA: Henry Holt & Company.

[9] J. Paiget, *The child's conception of number*. 1952, London: Routledge & Kegan Paul Ltd. 258.

[10] L.P. Rieber and M.J. Hannafin, *Effects of textual and animated orienting activities and practice on learning from computer-based instruction.* Computers in the Schools, 1988. **5**(1-2): p. 77-89.

[11] G.G. Robertson, J.D. Mackinlay, and S.K. Card, *Cone Trees: animated 3D visualizations of hierarchical information* in *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology* 1991 ACM Press: New Orleans, Louisiana, United States p. 189-194

[12] J. Stasko and E. Zhang, *Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations*, in *Proceedings of the IEEE Symposium on Information Visualization 2000.* 2000, IEEE Computer Society. p. 57.

[13] B. Tversky, J. Zacks, P.U. Lee, and J. Heiser, *Line, blobs, crosses and arrows: Diagrammatic communication with schematic figures*, in *Theory and Application of Diagrams*, Anderson, M., Cheng, P., and Haarslev, V., Editors. 2000, Springer: Berlin. p. 221-230.

[14] C. Ware, *Information Visualization : Perception for Design*. 2 ed. 2004: Morgan Kaufmann Publishers. 435.

[15] C. Ware and R. Bobrow, *Motion to support rapid interactive queries on node-link diagrams.* ACM Transactions on Applied Perception, 2004. **1**(1): p. 3-18.

[16] C. Ware, E. Neufeld, and L. Bartram. *Visualizing Causal Relations*. in *IEEE Information Visualization*. 1999. California, CA.