

What is a Visual Language?

Martin Erwig^a, Karl Smeltzer^a, Xiangyu Wang^a

^a*School of EECS, Oregon State University*

Abstract

The visual language research community does not have a single, universally agreed-upon definition of exactly what a visual language is. This is surprising since the field of visual languages has been a vibrant research area for over three decades now. Disagreement about fundamental definitions can undermine a field, fragment the research community, and potentially harm the research progress. To address this issue we have analyzed two decades of VL/HCC publications to clarify the concept of “visual language” as accepted by the VL/HCC community, effectively adopting the approach of descriptive linguistics. As a result we have obtained a succinct visual language ontology, which captures the essential aspects of visual languages and which can be used to characterize individual languages through visual language profiles. These profiles can tell whether and in what sense a notation can be considered a visual language. We also report trends from and statistics about the field of visual languages.

1. Introduction

S.K. Chang is one of the pioneers of the area of visual languages. He has edited and written some of the early books in this area [1, 2], at a time when the field was still in its infancy. Each scientific discipline must provide an answer to the question of what it is about, and consequently these books provided a definition of what a visual language is. However, while these early definitions were appropriate at the time and for the purpose at hand, they naturally were given from a specific vantage point and thus emphasize some aspects more than others. In the years since, several other definitions have been proposed (to be discussed in Section 2), but it may come as a surprise that within the visual language research community today, there does not appear to be any singular, accepted definition of exactly what constitutes a visual language. This is troublesome for a number of reasons.

First, competing definitions of visual languages can lead to misunderstandings when people employ different definitions. Softening existing definitions and removing prescriptive requirements could improve this situation. Second, researchers with a relatively narrow view of what constitutes a visual language may not be taking full advantage of the research tools and techniques available to them. For example, researchers

Email addresses: erwig@oregonstate.edu (Martin Erwig), smeltzek@oregonstate.edu (Karl Smeltzer), wangx4@oregonstate.edu (Xiangyu Wang)

in human-computer interaction (HCI) may be performing some particular research which is closely related to visual languages. If this is the case, then those researchers could complement their existing approach with visual language techniques. For instance, usability evaluations could potentially be strengthened by also evaluating notions of completeness and expressiveness. Third, a limited or narrow definition of a visual language might limit researchers who contemplate the development of a visual notation. And finally, it is conceivable that a lack of agreement among researchers about what is or is not a visual language could conceivably lead to a situation in which important research is missed by a suitable publication venue, because reviewers and authors assume different definitions.

Intriguingly, S.K. Chang's own early definition foretold the difficulty in establishing one single definition of visual languages [2]:

The term "visual language" means different things to different people.

Let us consider some specific examples of what might or might not be considered a visual language. In information and scientific visualization the general goal is, loosely, to encode raw data into a systematic, visual representation. Given this, it could then be argued that visualization is a visual language for communicating about data. It is not immediately clear, however, how far this overlap between visualization and visual languages extends. Should research on visualization tooling also be considered visual language research? Are visualization evaluation techniques immediately appropriate for evaluating visual languages of all kinds? While the reader may feel strongly one way or the other about these questions, there is no clear answer in the literature.

Next consider research on graphs. Most researchers would likely agree that graphs are a form of visual language, but it is not clear that all graph-related topics are relevant to general visual language researchers. For example, is graph drawing a visual language topic? It could be argued that it is, because it is similar to what pretty printing does for textual language. But what about graph interaction techniques or graph query languages? Even though a graph query language may itself be textual, it still relates to a visual language.

We believe that a working definition of visual languages together with a classification of visual language research is beneficial to both researchers in the field and to the field itself. It helps to set clear expectations and strengthens the communication among researchers.

At a high level, there are two possible approaches to arriving at an appropriate definition of the term "visual language". First, we could follow a *prescriptive approach* to craft a particular definition and then argue for its merits. This approach is attractive since it could deliver a crisp definition and provide clear guidance to the research community. However, the task also seems quite difficult, if not impossible, because it requires broad buy-in and agreement from across the research community. Alternatively, a *descriptive approach* would instead try to distill a definition from the existing work on the subject and thus essentially crystallize a definition the research community has already implicitly defined. That is, we could examine a representative sample of the research published in the field and try to identify common trends across the work that researchers and reviewers have already deemed appropriate for publication. The advantage of this approach is that

it does not require much arguing since it (a) more or less describes a state of affairs and (b) reflects the de-facto consensus the research community has reached. This approach is also sensitive to the potential evolution of the concept over time and can emphasize the validity of different meanings in different contexts.

This dichotomy of approaches is similar to how linguists approach the question of defining the English language. The prescriptive approach tries to mandate specific grammar and punctuation rules, whereas the descriptive approach acknowledges English as an evolving language.¹ We have elected to pursue the descriptive approach, and have accordingly surveyed 796 papers that were published in the last twenty years as part of the VL/HCC symposium.² We have selected this conference since it is the premier, fast-turnaround publication venue for visual languages and thus provides an accurate view into the state of research in this area. (We will say more about our methodology in Section 3.)

In this work, we make the following specific contributions. First, we provide an overview of what kinds of research has been accepted by the VL community over the past two decades. By systematically analyzing the publications we can distill the most salient features of visual languages as considered by the VL community as a whole. This will result in a *light-weight ontology* of visual languages, described in Section 4, which can be used to characterize any visual language by assigning it a *visual language profile* that locates it in the space defined by the ontology. The ontology can also be used as a domain-specific language for querying visual language profiles. In fact, we have effectively done that when determining some of the statistics and figures for this paper.

Second, we explore whether, and if so, how the concept of visual languages has evolved over time. As it turns out, while the number of papers on visual languages has somewhat declined over the years, visual languages constitute a significant part of the VL/HCC symposium. Moreover, the types of languages investigated have remained relatively constant. This fact allows us to provide a characterization of the concept of “visual language” that is quite stable over time.

Finally, by providing insight into what work might be classified as being related to visual languages, we provide some guidance to researchers on how they can complement their current research tools and techniques by adding those already established within the visual language community.

The remainder of this paper is structured as follows. In Section 2 we review some definitions of visual languages, given by different researchers. While these definitions agree on some aspects, they differ on others, which reflects the limitations of a prescriptive approach. In Section 3 we explain the methodology of our analysis and provide a brief overview of the data set that forms the basis for our investigation. Section 4 describes in some detail the ontology that has resulted from our analysis of the data. We provide examples and illustrate how to employ visual languages profiles to characterize individual visual languages. In Section 5 we present the results of our

¹See the two articles by Nunberg [3] and the reply by Halpern [4] to get an impression of this ongoing debate in the linguistic community.

²The complete data set is available at the following URL: eecs.oregonstate.edu/~erwig/vlhcc-paper-classification-1995-2014.csv

analysis, a quantitative assessment of the different kinds of visual languages that were researched over the years. We note threats to validity in Section 6, discuss related work in Section 7, and present conclusions in Section 8.

2. Definitions of Visual Languages

With the significant amount of work already available in the field of visual languages, several definitions have been proposed, particularly in books designed at least in part to serve as competent introductions to the field as a whole.

For instance, Marriott and Meyer include the following definition in their collection of visual language work [5].

By a visual language we mean a set of diagrams which are valid “sentences” in that languages where a diagram is a collection of “symbols” in a two or three dimensional space. Which sentences are valid and what their meaning is depends on the spatial relationships between the symbols. Thus, for example, mathematical expressions, plans, and musical notation are commonly used visual languages.

This view is mirrored in the definition given by Bottoni et al. [6], in which visual languages are defined in terms of visual sentences. However, their view is arguably more general since they do not require sentences to consist of symbols that are related to one another.

The theory of visual sentences formalizes the way the computer associates a computation meaning with an image shown on the computer screen and, conversely, the way it generates an image on the screen from a computation. The visual sentence is defined as an interpreted image and a visual language is viewed as a set of visual sentences in a user-computer dialogue.

Our view is similar to the previous two and emphasizes the difference between visual and textual languages as expressed in [7].

A textual language is a set of strings over an alphabet. The symbols of any sentence are only related to each other by a linear ordering. In contrast, a sentence of a visual language consists of a set of symbols that are, in general, related by several relationships.

Those definitions focus mainly on the formal view of visual languages. Others propose even broader definitions. Zhang [8], for example, defines the notion of a visual language as follows.

A pictorial representation of conceptual entities and operations and is essentially a tool through which users compose visual sentences. [...] In a broader sense, visual languages refer to any kinds [of] non-textual but visible human communication medias, including art, images, sign languages, maps, and charts, to name a few.

While there are certainly commonalities among these definitions, they differ in their scope. Any attempt on our part to propose yet another such definition is likely to either add layers of confusion and nuance to an already complex subject or to generalize it to the point of a truism. For example, Horn gives the following, very high-level definition [9] that does not provide much of an explanation.

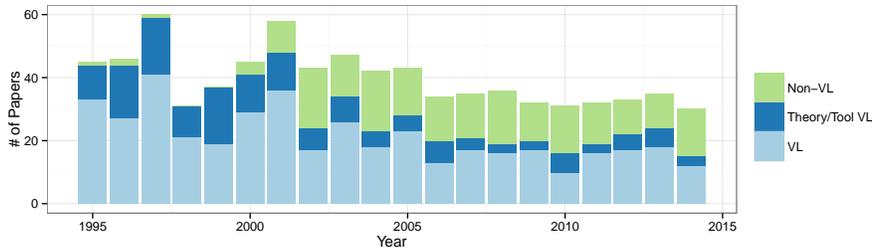


Figure 1: Trend of VL/HCC publications. Over the last 20 years the number of published papers went down. For the last 10 years, the ratio between papers about visual languages and other papers remained relatively constant.

The full integration of words, images, and shapes into a single, unified communication unit is visual language.

Finally, Chang et al. have distinguished early on between different interpretations of the term “visual language” [2].

The term “visual language” means different things to different people. To some, it means the objects handled by the language are visual. To others, it means the language itself is visual. To the first group of people, “visual language” means “language for processing visual information,” or “visual information processing language.” To the second group of people, “visual language” means “language for programming with visual expressions,” or “visual programming language.”

This characterization is remarkable prescient as it recognizes the difficulty in providing one single, focused definition of what a visual language is.

None of the above definitions could be objectively judged as correct or incorrect or even better or worse than the others. The definitions reflect the insights and perspectives of the researchers at a particular point in time. As the data we collected demonstrate, these perceptions can change over time, as can the relevance of specific view points. Therefore, while prescriptive definitions provide some insights into the nature of visual languages, a descriptive characterization offers a more flexible and adaptive characterization of the field.

3. Methodology and Overview

We started our descriptive approach to a definition of visual languages by surveying 20 years of papers published at VL/HCC and its previous incarnations. Specifically, we cataloged all publications from VL/HCC for the years 2004–2014, HCC for the years 2001–2003, and VL for the years 1995–2000. Altogether, this provided us a corpus of 796 papers.

We thoroughly read the abstract and introduction for every paper, and at least skimmed every other section. For many papers, individual sections were read in greater

detail as necessary. Initially, each paper was examined one author, but any which lead to uncertainty were examined and discussed by all authors.

Each paper was first categorized by whether or not it was in any way related to visual languages. In this initial assessment we granted ourselves extreme latitude and only excluded papers from the visual language category which were objectively and obviously about a different topic. This was to protect the selection process from a potential initial bias and preconceived notions about what constitutes a visual language. This process left us with 594 visual language papers and 202 papers on other topics.

Critically, this does not suggest that we relied on a prescriptive definition of visual languages to perform this task. Instead, we erred on the side of including all potentially relevant papers and then depend on the forthcoming analysis of the data to provide our descriptive definition.

To provide some specific examples of this initial classification, we considered work such as Buono et al. [10] and Hancock [11] to be visual language papers although they do not principally propose or describe a language of manipulable visual elements, but rather only include elements which may be considered a form of visual communication. Kline et al. [12] and Davidson et al. [13] illustrate cases which are objectively not about visual languages, instead focusing on aspects of human-computer interaction and traditional text-based software development.

Each paper deemed to be about visual languages was then further classified into one of three categories as follows. If the paper was about one (or more) specific visual language(s), we applied the ontology described in Section 4 to it. Otherwise, if the paper was about visual languages in general, we classified it as either a theory or a tool paper. Note that the ontology for classifying visual languages was not given a priori, but rather evolved during as part of the classification process (more details on that in Section 4). Figure 1 shows a breakdown of the paper corpus by year, separated into categories for visual language papers, tool and theory visual language papers, and papers on other topics.

As Figure 1 shows, the relative number of papers about visual languages proper declined over the years while the number of papers about other subjects increased. This reflects a trend in the history of the VL/HCC conference that is not very surprising for people in the community. Ultimately, 594 papers about visual languages were published over the last two decades in VL/HCC. The analysis of the languages follows in Section 5. But first we will describe in the next section the visual language ontology that we used for that purpose.

4. Visual Language Ontology

One outcome of our analysis of VL/HCC publications is an ontology of visual languages. This ontology is given by a collection of tags that may have additional attributes. The tags are grouped into two major categories to describe different aspects of a visual language, explained in detail below. The ontology serves three purposes. First, it provides a high-level summary of the research field and thus gives a direct answer to the question raised by this paper. Second, it can be employed to characterize individual visual languages, and third, it contributes a structure for the analysis of trends within the field.

This ontology was itself developed alongside the analysis of the VL/HCC papers. We started with an initial collection of tags and attributes that we tried to apply to the languages that we encountered. This draft was then amended whenever we were confronted with languages that did not fit the ontology and required new or different tags or attributes. We also have removed some tags from our initial draft that were never used. Thus, this ontology is not a prescriptive view, but a descriptive schema that was discovered and has evolved as part of the data analysis. Since this process was iterative, every paper was considered multiple times to ensure that any new tags and attributes were assigned where appropriate.

The tags are grouped into two major categories to characterize the syntactic appearance and the semantics of the notation. The tags within each category are not mutually exclusive, as languages can combine multiple notations and semantic aspects (examples are given below).

4.1. Syntactic Appearance

For the concrete syntax we found that visual languages exhibit basically four major kinds of syntactic features, summarized below. The additional tags 1D, 2D, 3D, CONTINUOUS, and RECURSIVE denote obvious syntactic language features which we consider minor aspects.

Syntactic Structure
GRAPH[<i>Directed, UnDirected, [Node Edge]Labeled</i>]
PARTITION[<i>Open, Closed</i>]
ICON
TEXT[<i>Plain, Structured</i>]
1D, 2D, 3D, CONTINUOUS, RECURSIVE

The first two tags tell how a language makes use of space: A GRAPH language uses explicit nodes that are connected by edges. Edges can be directed or undirected, and nodes as well as edges can be labeled. In their purest form nodes and edges do not occupy significant amounts of space and could in principle be made arbitrarily small/thin. The space that is not occupied by nodes, edges, or labels belongs to the background and carries no meaning. In contrast, a PARTITION language divides the space into (non-overlapping) regions. A closed partition does so completely and has no background, whereas an open partition does leave unused space as background.

The next two tags provide more details about the structure of the visual language. First, notations in which icons play an important role are tagged as ICON. Note that we did not include visual languages in this category that only used pictures to distinguish between different node types, which happened quite often. Second, the TEXT tag is used for languages that employ plain labels or structured expressions (that are defined by a grammar and define a sublanguage) in addition to visual notation.

Finally, we have several tags to characterize important but less fundamental notation features. First, languages are tagged according to their dimensionality with 1D, 2D, and 3D. Also, while the visual notation of most languages represents discrete objects, there are some that use CONTINUOUS displays. Lastly when the syntactic structure can be applied on multiple levels, we tag the language as RECURSIVE. This happens, for

example, when nodes of a graph can contain other graphs or when cells of a table can contain other tables. In many cases a RECURSIVE language produces hierarchically structured visual programs, but that doesn't always have to be the case. For example, when a RECURSIVE GRAPH language allows edges from nodes in nested graphs to nodes on a higher level, this allows non-hierarchical visual programs to be constructed.

The attributes attached to some tags can be used to provide more details and to distinguish between notations on a more fine-grained level. The meaning of most attributes is rather obvious. In the following, we present some examples to illustrate the meaning of the tags and attributes.

GRAPH languages are wide-spread and have been used for a diverse set of applications.³

GRAPH[. . .]	
<i>Directed, Labeled</i>	LabView [14]
<i>Directed, NodeLabeled</i>	Neuron Diagrams [15]
<i>Directed, EdgeLabeled</i>	Abstract Syntax Graphs [16]
<i>UnDirected, Labeled</i>	ER class diagrams [17]
<i>UnDirected, NodeLabeled</i>	VEX [18]
<i>UnDirected, EdgeLabeled</i>	Euler Graphs [19]

The two most prominent examples of PARTITION languages are probably spreadsheets and Euler diagrams. In addition to the *Open/Closed* difference in how the underlying space is employed, they also differ in the way they make use of text: Euler Diagrams simply use *Plain* labels, whereas spreadsheets use formulas, which are *Structured* and defined by a grammar.

PARTITION[. . .]	
<i>Open</i>	Euler Diagrams [20]
<i>Closed</i>	Spreadsheets

While GRAPH and PARTITION languages are quite different in nature, realizing that they use space in opposite ways makes it unsurprising that some languages combine both representational principles.

GRAPH[. . .] & PARTITION[. . .]	
<i>NodeLabeled & Open</i>	Spider Diagrams [21]
<i>NodeLabeled & Closed</i>	Probula [22]

The classification as GRAPH & PARTITION does not capture some of the differences between them that are in the concrete way of how they are combined. For example, while Spider Diagrams place nodes inside of regions of a partition, Probula makes (sub)partitions into nodes that are connected by edges.

³In the following, we have tried to use citations of VL/HCC papers as references whenever possible and appropriate. For some languages, however, we have chosen either the original or a less specialized publication, which may have occurred in a different venue.

This observation reveals a limitation in expressiveness of the ontology, but, more importantly, it illustrates a trade-off in the design of the ontology, namely between simplicity and expressiveness. Specifically, to reflect the difference in the ontology, we would need more tags or, worse, an additional linguistic mechanism for talking about combinations of tags. While a highly expressive ontology can achieve more precise characterizations of visual languages, it is more difficult to understand and use. Since the main purpose of the ontology in this work is not the exact characterization of each individual visual language, but rather the description of the visual language landscape, simplicity of the ontology is an overriding goal.

In a RECURSIVE language the underlying spatial principle can be applied in some nested way. By “spatial principle” we mean the way syntactic appearance of the language is structured. It applies to only GRAPH and PARTITION languages.

... & RECURSIVE	
GRAPH[<i>UnDirected, NodeLabeled</i>]	VEX [18]
GRAPH[<i>Directed, Labeled</i>]	LabView [14]
PARTITION[<i>Open</i>]	Euler Diagrams [20]
PARTITION[<i>Closed</i>]	SGViewer [23]

Recursive partitions can also appear in combination with graphs, as evidenced by Spider Diagrams [21] or the visual XML query language Xing [24].

4.2. Semantics

The salient features of a visual language are contained in its appearance, but the essence of a visual language also includes its semantics. Therefore, we have a category of tags for characterizing semantic domains, which capture the semantics in a very abstract and high level. (We distinguish tags for semantic domains from tags for syntactic appearance by a trailing *.)

Semantic Domain
STATIC*
DYNAMIC*
GRAPHICAL*
EXTERNAL*

Sentences of a DYNAMIC* visual language denote a computation, which is a transformation of some representation. For example, a finite automaton is a visual description of a function that maps strings to booleans (accepted strings are mapped to true, and rejected strings are mapped to false). In terms of denotational semantics this means that the semantics domain of the language is a function [25, 26]. In the case of the finite automaton the semantic domain is $String \rightarrow Bool$. In contrast, sentences of a STATIC* visual language denote some fixed structure. For example, the semantics of a visual ontology language such as VOWL [27] is a set of objects and relationships and not a computation. Another distinctive aspect is whether the semantic domain consists of visual elements itself, which can be the case for DYNAMIC* as well as for STATIC* languages. An EXTERNAL* semantic domain is distinct from the visual notation. It can be itself a (different) GRAPHICAL* notation or some textual language.

An ER or UML class diagram denotes a DB schema. Since no transformation is involved, these are *STATIC** visual languages. The schema is a mathematical description of relations and attributes that does not contain diamonds, ovals, lines, etc. The domain is therefore *EXTERNAL** to the notation. A notation whose semantic domain is *STATIC** and *GRAPHICAL** (but not *EXTERNAL**) is given by Visual Graphs [28]. In contrast, VAS [29] is a visual notation for other visual languages and thus is *STATIC* & GRAPHICAL* & EXTERNAL**.

<i>STATIC* & . . .</i>	
<i>EXTERNAL*</i>	ER/UML class diagrams [30]
<i>GRAPHICAL*</i>	Visual Graphs [28]
<i>GRAPHICAL* & EXTERNAL*</i>	VAS [29]

Many *DYNAMIC** languages can be also classified as *STATIC** languages. Consider, for example, Euler Diagrams [20], which are *DYNAMIC** since they denote predicates (that is, boolean functions) on sets, but they could also be classified as *STATIC** since they denote statements in propositional logic. Since we can always find a separate *STATIC** representation for an otherwise *DYNAMIC** semantic domain, at least for *EXTERNAL** ones, we have classified those cases as *DYNAMIC** since this is the ultimate semantics.

Two prime examples of *DYNAMIC** languages are finite automata, which denote predicates on strings and thus have an *EXTERNAL** semantic domain, and AgentSheets [31], which denotes animations. Similar to the overlap between *STATIC** and *DYNAMIC** languages, there is an overlap between *GRAPHICAL** (and non-*EXTERNAL**) and *EXTERNAL** (and non-*GRAPHICAL**) languages. A prime example is VEX [18], which is a visual representation of lambda calculus. Its semantics can be interpreted as rewrite rules on the the visual notation (that is, *GRAPHICAL**) or as the standard semantics for lambda calculus (that is, *EXTERNAL**). Since the lambda calculus semantics proper is not part of the visual language definition, the non-*GRAPHICAL** interpretation of VEX's domain is also classified as *STATIC** (similar to Euler Diagrams). Again, since the dynamic semantics interpretation weighs stronger, we categorize VEX therefore as *DYNAMIC* & GRAPHICAL** and not as *STATIC* & EXTERNAL**.

<i>DYNAMIC* & . . .</i>	
<i>EXTERNAL*</i>	Finite automata
<i>GRAPHICAL*</i>	AgentSheets [31]
<i>GRAPHICAL*</i>	VEX [18]
<i>GRAPHICAL* & EXTERNAL*</i>	CONVERt [32]

We could have also used the application domain for classification, but decided against it. While this information is certainly useful, it does not say much about the visual language proper. Again, the purpose of the ontology is not a comprehensive and detailed classification of visual languages, but to distill the major distinguishing features of visual languages. For the same reason we did not distinguish between visual programming languages, query languages, specification languages, etc.

The visual languages mentioned so far have illustrated the most important aspects of the ontology. While we cannot, for lack of space, give an example for every profile that

we found, we do want to mention a few interesting, and maybe unexpected, cases. An example for an ICON & 1D language is QueryMarvel [33] for expressing temporal patterns. Its semantic domain is EXTERNAL* & DYNAMIC*. We have also seen a language that combines a 1D & ICON notation and situates it in a 3D context to express spatio-temporal patterns [23]. Visual languages with a CONTINUOUS notation have been mostly used as parts of other languages or systems to model GUI elements such as sliders [34] or for indicating animation traces [35].

4.3. Derived Tags & Visual Language Profiles

Based on the ontology, each visual language can be assigned a *profile*. The tags and their categories define a design space for visual languages, and each profile acts as a “vector” that locates it in this space.

From the structure of the ontology as a collection of tags it follows that many languages share subsets of their tags. Thus, to obtain more succinct descriptions of certain classes of visual languages it is helpful to expand the ontology by derived tags. Some derived tags identify a particular visual language category. In such cases, the derived tag effectively serves as a definition of such a category.

For example, the most widely used interpretation a dataflow language is as follows. Since the ontology does not allow for the semantic distinction between data and control flow, we use the more general derived tag FLOW, which covers both flow-based language categories.

$$\text{FLOW} = \text{GRAPH}[\textit{Directed}, \textit{NodeLabeled}] \& \text{DYNAMIC*} \& \text{EXTERNAL*}$$

However, the (data)flow paradigm has been creatively adapted in several ways. For example, the semantic domain in Tanimoto’s data factory [36] is not EXTERNAL*, but rather employs the concrete program representation and is thus also GRAPHICAL*. If we adopt the convention that tags can be overwritten, then Tanimoto’s data factory could be classified as FLOW & GRAPHICAL*. Another example is Envision, a visual programming environment [37], which combines dataflow with PARTITION[Open] & RECURSIVE to manage large object-oriented programs.

A derived tag effectively specifies a subspace of all visual languages. Adding tags thus corresponds to an intersection operation on the spaces associated with the tags. This allows for gradual refinements of categories. An example is given by spreadsheets, which can be incrementally defined as follows (only characterizing the syntactic aspects).

$$\begin{aligned} \text{TABULAR} &= \text{PARTITION}[\textit{Closed}] \\ \text{FORMULA} &= \text{TEXT}[\textit{Structured}] \\ \text{SPREADSHEET} &= \text{TABULAR} \& \text{FORMULA} \end{aligned}$$

Derived categories for different languages that build on one another also effectively describe subclasses of visual languages, as in the case of Euler diagrams and Spider diagrams. We are using the following abbreviations.

$$\begin{aligned} \text{LABEL} &= \text{TEXT}[\textit{Plain}] \\ \text{CURVED} &= \text{PARTITION}[\textit{Open}] \end{aligned}$$

Here is the definition of one language as a subclass of another.

Euler = CURVED & LABEL & EXTERNAL* & DYNAMIC*
Spider = Euler & GRAPH[*Directed, Labeled*]

Since derived categories can be used in the definition of other derived categories, we obtain a small domain-specific language for describing visual languages. Derived categories also provide further insights into what kind of visual languages have been defined and studied by the research community, and we will use some derived categories in the data analysis that follows.

With the basic tags from the ontology and the derived tags we can assign a profile to each visual language that identifies it in the space defined by the ontology. We can nicely reuse the derived categories. Here are some prominent examples.

LabView : FLOW & FORMULA & RECURSIVE
Excel : SPREADSHEET & GRAPHICAL* & STATIC*

The semantics of a spreadsheet is another spreadsheet in which all formulas have been replaced by their resulting values. The semantics domain is thus GRAPHICAL* (and not EXTERNAL*). It is also STATIC* since the semantics of a spreadsheet is not a computation, but another spreadsheet (without formulas). The difference to other DYNAMIC* languages is that the semantics of a spreadsheet does not take any input for a further computation; the input is already part of the spreadsheet and gets processed as part of its semantics.

Combinations of tags can be immediately employed to express queries on a collection of visual languages with associated profiles. In fact, we have used profiles in this way in the process of analyzing the data we gathered from the publications. We will present and discuss the results in the following section.

It is worth noting that in the context of these profiles no assumptions are made about tags that are absent. That is, a profile which does not contain a particular tag does not require that tag to be absent, but rather just accepts either case.

4.4. Tag Summary

In this section, we have introduced a total of 9 syntactic tags and 4 semantic tags as well as attributes for some of the tags which allow them to be further refined. Furthermore, sets of tags can be assembled to define derived tags, which define subspaces of visual languages. A more complete example of paper classification is shown in table 1. Represented papers were chosen arbitrarily from the 2014 VL/HCC conference.

The presented ontology, together with its application in the following section, is what provides a descriptive definition of visual languages.

5. The Landscape of Visual Language Research

With our ontology, we can begin to explore the corpus of VL/HCC papers. This section describes some of the interesting rates, overlaps, and trends regarding the occurrence of ontology features. In accordance with our descriptive approach, these statistics help to explain what features are likely to occur in any particular work on visual languages.

	Tag	[38]	[39]	[40]	[41]
Syntactic Appearance	GRAPH	✓	✓		
	<i>Directed</i>		✓		
	<i>UnDirected</i>	✓			
	<i>NodeLabeled</i>	✓	✓		
	<i>EdgeLabeled</i>		✓		
	PARTITION			✓	✓
	<i>Open</i>			✓	✓
	<i>Closed</i>			✓	✓
Syntactic Features	ICON	✓	✓		
	TEXT	✓	✓	✓	✓
	<i>Plain</i>	✓	✓		✓
	<i>Structured</i>			✓	✓
	1D				
	2D	✓	✓	✓	✓
	3D				
	CONTINUOUS RECURSIVE				
Semantics	STATIC*		✓		✓
	DYNAMIC*	✓		✓	
	GRAPHICAL*				✓
	EXTERNAL*	✓	✓	✓	

Table 1: Example classifications

At the highest level, of the 594 visual language papers, 430 were focused on describing a specific language; the rest consisted of 102 theory papers and 62 tool papers where 30 of the theory papers and 24 of the tool papers could be assigned a specific visual language profile. Thus, a total of 484 papers could be included in the data analysis of visual languages.

5.1. Prominent Syntactic Features

The vast majority of visual languages are either GRAPH (36%) or PARTITION (23%) or both (26%). Other visual languages (15%) include those that primarily employ icons (2%) or use an iconic representation together with graphs or partitions (6%). This goes to show that the two primary approaches to utilize space for visual languages are GRAPH and PARTITION, which is reflected by the fact that an overwhelming number of papers (91%) make use of the strengths of one or more of these approaches.

As can be seen in Figure 2, over time, the number of visual languages that are neither GRAPH nor PARTITION appears to decrease. However, there is no clear trend as to whether GRAPH or PARTITION dominates, which again indicates that these two are somewhat orthogonal visual language design principles.

Table 2 shows the most frequent visual language types and concrete visual languages, as defined in Section 4, that we found in the data set. Note that these are mutually

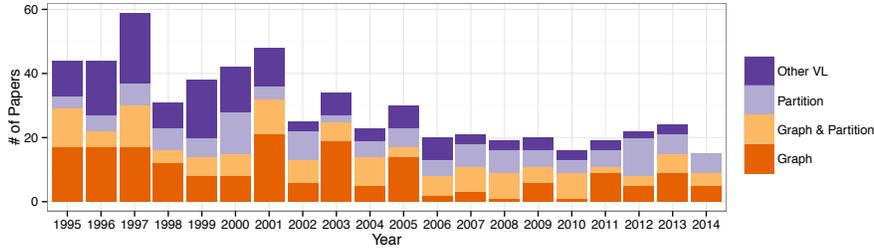


Figure 2: Trend of major visual language principles. GRAPH and PARTITION languages clearly dominate the field. Other types of visual languages have declined over the years.

exclusive, that is, the TABULAR row does not include any of the SPREADSHEET papers, and the Euler Diagram row does not include any Spider Diagram papers. Note also that the Tool and Theory columns contain numbers for those papers that describe tools or theoretical aspects for the specific (kinds) of visual language. Those papers, for the most part, do not present or discuss language designs or extensions, but focus on particular tool/theory aspects.

	Tool	Theory	Specific VL	Total
FLOW	9	10	179	198
SPREADSHEET	1	6	66	73
TABULAR	2	2	43	47
Spider	2	5	24	31
Euler	2	0	18	20

Table 2: Most popular visual languages

We can already begin to see the usefulness of our approach of using the ontology as a query language, as queries provide insight and raise further questions. For example, we observe that the majority (73 out of 120) of all papers about tabular visual languages are focused primarily on spreadsheets. This has prompted us to investigate those 47 remaining papers even further. Indeed, in doing so we see that the key differentiating aspect between spreadsheets and other tabular languages is in the nature of their semantic domain. The semantic domain of all 47 non-spreadsheet tabular languages is EXTERNAL*, that is, the table aspect is itself not part of the semantic domain. In the spreadsheet languages, however, the semantic domain is itself a spreadsheet.

A full 84% of the visual languages categorized using our ontology contained some form of textual content. Out of those which did contain text, 42% were *Structured* and the remainder were *Plain*. Due again to their relative popularity, spreadsheet languages accounted for 41% of the *Structured* text. Neither type of text showed any obvious trends over time, both aligning with the general trend in visual languages as a whole.

5.2. Prominent Semantics Features

Table 3 shows a breakdown of all the papers classified in our ontology according to their semantic domains.

	EXTERNAL*	¬EXTERNAL*
GRAPHICAL*	35	102
¬GRAPHICAL*	347	0

Table 3: Breakdown of the semantic domains.

The first item of note is the lack of papers which are neither classified as EXTERNAL* nor as GRAPHICAL*. This is not surprising. By definition, if the semantic domain of a visual language is not EXTERNAL*, then it must be made up of the same visual notation used in the language itself. Therefore, any semantic domain which is not EXTERNAL*, must be GRAPHICAL*. Note that papers which are classified as tool or theory papers and do not contain a specific visual language cannot be classified in this table since they do not allow the identification of a single semantic domain. Thus, those papers are omitted here.

Next we can see that 347 languages are classified as EXTERNAL* only (that is, they are not also GRAPHICAL*). This tells us that the majority of the visual languages discussed in this corpus have semantic domains which are mathematical models, computations, textual languages, and so on.

Relatively few papers are tagged as both GRAPHICAL* and EXTERNAL*, that is, papers that denote visual objects that are not part of their own notation. This category contains a substantial number of works on data, software, and algorithm visualization. It is not surprising, then, that out of the 35 papers categorized here, 22 of them are also tagged GRAPH.

The fourth category contains the work tagged only as GRAPHICAL*. This category is noticeably larger than the previous category, in part because it contains all the work on spreadsheets. In fact, spreadsheet papers account for more than two-thirds of this category (73 out of 102).

Finally, we examined whether or not the semantic domain classifications seem to have any temporal relationships. Two of the three non-empty categories roughly follow the same trend as visual language papers overall. However, the work tagged only as GRAPHICAL* appears to break from this trend, appearing much flatter or perhaps even a very slight upward trend. One possible explanation for this is that even as the research has broadened in scope to include more human-computer interaction and end-user programming research, spreadsheets have stayed extremely relevant and popular.

We also explored the relationship between languages with a DYNAMIC* semantic domain (340) and those with a STATIC* domain (144). Out of 20 years, 19 produced more papers tagged DYNAMIC* than STATIC*. The sole exception came in 2002, where we saw 10 DYNAMIC* tags and 12 STATIC* tags.

When charted over time, it was immediately apparent that DYNAMIC* essentially followed the overall trend of visual languages, decreasing somewhat over time. Interestingly, however, STATIC* gave a more bimodal shape with peaks during both early

and later years. We initially speculated that this could be caused by the popularity of spreadsheet research, as all such work is tagged `STATIC*`, but this turned out to be false. When all spreadsheet papers were removed, the `STATIC*` data still showed distinct peaks in the early and later data.

6. Threats to Validity

There are two main threats to the validity of our analysis: (1) the restriction to the VL/HCC conference series and (2) the initial classification of papers as visual languages (or not). Moreover, the distinction of semantics as `DYNAMIC*` or `STATIC*` is not always possible. This leaves some degree of freedom and allows for some ambiguity in the classification.

In our descriptive approach for what visual languages are we have focused exclusively on the VL/HCC conference series. However, work pertinent to visual languages has been published in other venues as well. Specifically, the *Journal of Visual Languages* (JVLC) and the biennial *Diagrams* conference series are outlets for presenting visual language research. Excluding publications from those two venues could lead to results that are too narrow and potentially biased.

We believe that our analysis is still valid and useful for the following reasons. First, our data set is already quite large and thus chances are that it covers most types of visual language research. Second, there is considerable overlap between JVLC and VL/HCC. Many papers in JVLC are long versions of papers that have previously been published in VL/HCC. In fact, many VL/HCC conferences have later republished their best papers in a special issue of JVLC. Third, the Diagrams conference has a much broader scope and covers areas such as psychology and philosophy. Also, many of the papers in Diagrams that would classify as visual languages are related to papers that also appeared in VL/HCC. In summary, we acknowledge that the descriptive definition of visual language in this work is restricted to the view of the VL/HCC community and might miss some aspects that could be found in papers from other venues, but we also believe that the large set of papers studied provides a representative sample and serves our purpose well. Moreover, in the spirit of a descriptive approach to visual language characterization and classification we do not consider the results reported in this paper as a final, conclusive judgment, but rather as an analysis that we expect to be expanded upon by other researchers in the future.

During our initial scan through the proceedings we had to decide for each paper whether it is about one or more concrete visual languages. Only in that case did we classify it according to the ontology and assign a profile to it. If this initial classification were too narrow, the observed results and trends would have been skewed. To address this concern, we adopted an include-by-default policy: Every paper was considered a potential visual language paper and was excluded only if it was clearly not. This occurred when the paper explicitly claimed to belong to a different field or if it was impossible to assign any profile to the notation used in the paper. We also discussed cases that were unclear until we reached a consensus. Finally, we make our data collection open and accessible to the public. This makes our analysis results transparent. This also helps other researchers to extend and broaden our analysis.

7. Related Work

One early system for classifying types of visual language papers came from Burnett and Baker [42], which was later expanded into the online “Visual Language Research Bibliography”.⁴ While effective, this work took a primarily prescriptive approach. As mentioned previously, a successful prescriptive approach requires buy-in and agreement from the community. By contrast, our descriptive approach relies only on what the community has already accepted. The Visual Language Research Bibliography seems to be not actively maintained anymore; the web site has been last updated in 2009.

Marriott and Meyer [43] proposed a hierarchy for visual languages roughly analogous to Chomsky’s textual language hierarchy, which was focused on the expressiveness and cost of parsing for various kinds of visual languages. Marriott and Meyer’s work is concerned with differentiating visual languages based on the underlying grammars.

Costagliola et al. developed a hierarchical system for categorizing the syntax of visual languages [44]. Despite sharing some common ideas with this work, it is proposed in a prescriptive style and small extensions require adding entirely new syntactic categories rather than attributes.

Hils provided a survey of visual dataflow languages [45]. While that survey groups languages based on their application domain, our ontology has deliberately refrained from doing that to better be able to focus on the question of what a visual language is. Hils’s survey emphasizes as one strength of dataflow languages that they have the potential to expand the appeal of visual programming by applying visual programming to new application domains and users associated with those domains.

Münch and Schürr surveyed visual languages used in industry and proposed some general improvements in such languages, in particular with regard to scalability [46].

Catarci et al. presented a survey of different visual query languages [47]. Similar to some of our syntactic tags, they distinguish between *form-based*, *diagram-based*, and *icon-based* representations, where diagram-based includes what we have classified as GRAPH. However, since their survey is limited to query languages, it cannot provide a full picture of the features employed by visual languages.

Finally, there are also some older surveys [48, 49], but those necessarily omit all of what has happened in the last 20 years, which is what this paper is focused on.

8. Conclusions

We have analyzed two decades of visual language research and thus addressed an issue that was raised by S.K. Chang over three decades ago, namely that different people mean different things by the term “visual language”. By employing a descriptive linguistics approach we have created a concise visual language ontology and have used it to capture the visual language essence of visual languages in 594 publications. This puts us in position to finally provide an answer to the question raised by the title of this paper. From our analysis, we conclude the following.

⁴<http://web.engr.oregonstate.edu/~burnett/vpl.html>

A visual language is a language whose syntactic structure can be classified as GRAPH, PARTITION, or ICON.

This still leaves open a wide range of possibilities. In particular, distinctions between different visual languages can be expressed on a more fine-grained level by employing other components of our ontology. And this is exactly the point of the descriptive approach that provides a spectrum of features that one can find in visual languages. As our analysis has demonstrated, some features are more prominent than others, and if a particular language has a prominent feature, it is more likely or strongly counted as a visual language than if it does not. We can see a similarity to this view in other fields. For example, instead of talking about exact positions of particles, physicists use probability distributions. In the same way, a visual language is—according to the descriptive approach—not defined by a crisp predicate, but by a distribution of frequencies of relevant tags that capture its prominent features.

Now what are prominent visual language features? As our investigation has shown, a visual language typically employs, according to the visual language community, either a graph-based or a partition-based notation, or a combination of both. At least 91% of all surveyed papers do. Most visual languages contain some form of text (84%), with majority employing plain labels (58% of the 84%). The semantics of visual languages are predominantly given by external domains (78%), which are mostly not GRAPHICAL* (92% of the 78%). Moreover, DYNAMIC* semantic domains are with 70% significantly more prevalent than STATIC* ones. A unifying theme among most of the surveyed visual languages is that they identify a systematic organizing principle for space, while the syntactic and semantic concepts they incorporate are mostly drawn from a small set of well-established and well-researched topics. Innovations often result from new tools, theoretical analyses, and extensions and combinations of features.

Finally, in addition to answering the question of what a visual language is, our investigation provides also some insights into their relevance. As the data show, although the number of visual language papers has gone down over the years—in their place other research areas have taken hold, they have reached a solid level. Of all papers surveyed over the twenty year period, 75% were about visual languages. Over the last ten years, the ratio is 60%, and there is no obvious downward trend noticeable. This shows that visual language research is still significant and thriving.

Acknowledgments

This work is partially supported by the National Science Foundation under the grants CCF-1219165 and IIS-1314384.

Bibliography

- [1] S.-K. Chang, K. S. Fu, Pictorial Information Systems, Springer Verlag, 1980.
- [2] S.-K. Chang, T. Ichikawa, P. A. Ligomenides, Visual Languages, Plenum Press, 1986.

- [3] G. Nunberg, The Decline of Grammar, *The Atlantic Monthly* 252 (6) (1983) 31–46.
- [4] M. Halpern, A War That Never Ends, *The Atlantic Monthly* 279 (3) (1999) 19–22.
- [5] K. Marriott, B. Meyer, K. B. Wittenburg, A survey of visual language specification and recognition, in: K. Marriott, B. Meyer (Eds.), *Visual Language Theory*, Springer-Verlag, 1998.
- [6] P. Bottoni, M. F. Costabile, S. Levialdi, P. Mussio, The theory of visual sentences to formalize interactive visual messages, in: F. Ferri (Ed.), *Visual Languages for Interactive Computing: Definitions and Formalizations*, IGI Global, 2008.
- [7] M. Erwig, Abstract Syntax and Semantics of Visual Languages, *Journal of Visual Languages and Computing* 9 (5) (1998) 461–483.
- [8] K. Zhang, *Visual Languages and Applications*, Springer-Verlag, 2007.
- [9] R. E. Horn, *Visual Language: Global Communication for the 21st Century*, MacroVU, Inc, 1998.
- [10] P. Buono, C. Ardito, M. F. Costabile, R. Lanzilotti, A. Piccinno, Dae: a visualization-based system for data analysis, in: *IEEE Int. Symp. on Visual Languages*, 2006, pp. 147–150.
- [11] C. Hancock, Toward a unified paradigm for constructing and understanding robot processes, in: *IEEE Int. Symp. on Human Centric Computing Languages and Environments*, 2002, pp. 107–109.
- [12] R. Kline, A. Seffah, H. Javahery, M. Donayee, J. Rilling, Quantifying developer experiences via heuristic and psychometric evaluation, in: *IEEE Int. Symp. on Human Centric Computing Languages and Environments*, 2002, pp. 34–36.
- [13] J. L. Davidson, R. Naik, U. A. Mannan, A. Azarbakht, C. Jensen, On older adults in free/open source software: reflections of contributors and community leaders, in: *IEEE Int. Symp. on Visual Languages and Human-Centric Computing*, 2014, pp. 93–100.
- [14] R. Jamal, L. Wenzel, The application of the Visual Programming Language LabVIEW to Large Real-World Applications, in: *IEEE Symp. on Visual Languages*, 1995, pp. 99–106.
- [15] M. Erwig, E. Walkingshaw, Causal Reasoning with Neuron Diagrams, in: *IEEE Int. Symp. on Visual Languages and Human-Centric Computing*, 2010, pp. 101–108.
- [16] M. Erwig, Semantics of Visual Languages, in: *13th IEEE Int. Symp. on Visual Languages*, 1997, pp. 304–311.
- [17] P. P.-S. Chen, The Entity-relationship Model—Toward a Unified View of Data, *ACM Trans. on Database Systems* 1 (1) (1976) 9–36.

- [18] W. Citrin, R. Hall, B. Zorn, Programming with Visual Expressions, in: 11th IEEE Symp. on Visual Languages, 1995, pp. 294–301.
- [19] P. Rodgers, G. Stapleton, J. Howse, Z. L., Euler Graph Transformations for Euler Diagram Layout, in: IEEE Symp. on Visual Languages and Human-Centric Computing, 2010, pp. 111–118.
- [20] L. Euler, Briefe an eine deutsche Prinzessin, aus dem Französischen übersetzt, 1773.
- [21] J. Howse, F. Molina, J. Taylor, S. Kent, Reasoning with Spider Diagrams, in: IEEE Symp. on Visual Languages, 1999, pp. 138–145.
- [22] M. Erwig, E. Walkingshaw, Visual Explanations of Probabilistic Reasoning, in: IEEE Int. Symp. on Visual Languages and Human-Centric Computing, 2009, pp. 23–27.
- [23] M. Sifer, O. Liechti, Zooming in One Dimension can be Better than Two: an Interface for Placing Search Results in Context with a Restricted Sitemap, in: IEEE Symp. on Visual Languages, 1999, pp. 72–79.
- [24] M. Erwig, A Visual Language for XML, in: 16th IEEE Int. Symp. on Visual Languages, 2000, pp. 47–54.
- [25] D. A. Schmidt, Denotational Semantics, Allyn and Bacon, Newton, MA, 1986.
- [26] J. C. Mitchell, Foundations for Programming Languages, 1998.
- [27] S. Lohmann, V. Link, E. Marbach, S. Negru, Webvowl: Web-based visualization of ontologies, in: Knowledge Engineering and Knowledge Management, 2015, pp. 154–158.
- [28] M. Erwig, Visual Graphs, in: 15th IEEE Int. Symp. on Visual Languages, 1999, pp. 122–129.
- [29] S. Üsküdarlı, T. B. Dinesh, The VAS Formalism in VASE, in: IEEE Symp. on Visual Languages, 1996, pp. 140–147.
- [30] J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Addison-Wesley Professional, 2004.
- [31] A. Repenning, T. Sumner, Agentsheets: A Medium for Creating Domain-Oriented Visual Languages, *Computer* 28 (3) (1995) 17–25.
- [32] I. Avazpour, J. Grundy, Using Concrete Visual Notations as First Class Citizens for Model Transformation Specification, in: IEEE Symp. on Visual Languages and Human-Centric Computing, 2013, pp. 87–90.
- [33] J. Jin, P. Szekely, QueryMarvel: A Visual Query Language for Temporal Patterns Using Comic Strips, in: IEEE Symp. on Visual Languages and Human-Centric Computing, 2009, pp. 207–214.

- [34] J. W. Atwood, M. M. Burnett, R. A. Walpole, E. M. Wilcox, S. Yang, Steering programs via time travel, in: IEEE Symp. on Visual Languages, 1996, pp. 4–11.
- [35] Y. Kato, E. Shibayama, Effect Lines for Specifying Animation Effects, in: IEEE Symp. on Visual Languages and Human-Centric Computing, 2004, pp. 27–34.
- [36] S. L. Tanimoto, Programming in a data factory, in: IEEE Symp. on Visual Languages and Human-Centric Computing, 2003, pp. 100–107.
- [37] D. Asenov, P. Müller, Envision: A fast and flexible visual code editor with fluid interactions (Overview), in: IEEE Symp. on Visual Languages and Human-Centric Computing, 2014, pp. 9–12.
- [38] J. Bresson, Reactive visual programs for computer-aided music composition, in: IEEE Int. Symp. on Visual Languages and Human-Centric Computing, 2014, pp. 141–144.
- [39] C. D. Schulze, R. Von Hanxleden, Automatic layout in the face of unattached comments, in: IEEE Int. Symp. on Visual Languages and Human-Centric Computing, 2014, pp. 41–44.
- [40] F. Turbak, D. Wolber, P. Medlock-Walton, The design of naming features in app inventor 2, in: IEEE Int. Symp. on Visual Languages and Human-Centric Computing, 2014, pp. 129–132.
- [41] K. S.-P. Chang, B. A. Myers, A spreadsheet model for using web service data, in: IEEE Int. Symp. on Visual Languages and Human-Centric Computing, 2014, pp. 169–176.
- [42] M. M. Burnett, M. J. Baker, A classification system for visual programming languages, *Journal of Visual Languages & Computing* 5 (3) (1994) 287–300.
- [43] K. Marriott, B. Meyer, Towards a hierarchy of visual languages, in: IEEE Int. Symp. on Visual Languages, 1996, pp. 196–203.
- [44] G. Costagliola, A. De Lucia, S. Orefice, G. Tortora, A framework of syntactic models for the implementation of visual languages, in: IEEE Int. Symp on Visual Languages, 1997, pp. 58–65.
- [45] D. Hils, Visual Languages and Computing Survey: Data Flow Visual Programming Languages, *Journal of Visual Languages and Computing* 3 (1) (1992) 69–101.
- [46] M. Münch, A. Schürr, Leaving the Visual Language Ghetto, in: IEEE Symp. on Visual Languages, 1999, pp. 148–155.
- [47] T. Catarci, M. F. Costabile, S. Levialdi, C. Batini, Visual Query Systems for Databases: A Survey, *Journal of Visual Languages and Computing* 8 (2) (1997) 215–260.
- [48] G. Raeder, A Survey of Current Graphical Programming Techniques, *Computer* 18 (8) (1985) 11–25.

- [49] B. A. Myers, Taxonomies of Visual Programming and Program Visualization, *Journal of Visual Languages and Computing* 1 (1) (1990) 97–123.