Journal of Visual Languages and Computing 25 (2014) 955–962

Contents lists available at ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

Short Paper

Local context-based recognition of sketched diagrams ☆

Gennaro Costagliola, Mattia De Rosa, Vittorio Fuccella *

Dipartimento di Informatica, University of Salerno, Via Giovanni Paolo II, 84084 Fisciano (SA), Italy

A R T I C L E I N F O

Article history:
Received 26 September 2014
Received in revised form
16 October 2014
Accepted 17 October 2014
Available online 29 October 2014

Keywords:
Local context
Sketch recognition
Multi-domain
Methodology
Framework
Visual languages

A B S T R A C T

We present a new methodology aimed at the design and implementation of a framework for sketch recognition enabling the recognition and interpretation of diagrams. The diagrams may contain different types of sketched graphic elements such as symbols, connectors, and text. Once symbols are distinguished from connectors and identified, the recognition proceeds by identifying the local context of each symbol. This is seen as the symbol interface exposed to the rest of the diagram and includes predefined attachment areas on each symbol. The definition of simple constraints on the local context of each symbol allows to greatly simplify the definition of the visual grammar, which is used only for further refinement and interpretation of the set of acceptable diagrams. We demonstrate the potential of the methodology using flowcharts and binary trees as examples.

1. Introduction

The use of diagrams is common in various disciplines. Typical examples include maps, line graphs, bar charts, engineering blueprints, architects' sketches, hand drawn schematics, etc. In general, diagrams can be created either by using pen and paper, or by using specific computer programs. Due to the unnaturality of WIMP interfaces, designers prefer to use pen and paper at the beginning of the design [39] and then transfer the diagram to the computer later [1].

To avoid this extra step, a solution is to allow users to sketch directly on the computer. This requires both specific hardware and sketch recognition based software. In terms of hardware, many pen/touch-based devices such as tablets, smartphones and interactive boards and tables are available today, also at reasonable costs. Sketch recognition is needed when the sketch must be processed and not considered as a simple image and it is crucial to the success of this new modality of interaction. It is a difficult problem due to the inherent imprecision and ambiguity of a freehand drawing and to the many domains of application.

In the literature, we can find the description of many different frameworks for the recognition of diagrams. There are both solutions developed for specific domains [24,18,34] and multi-domain solutions [1,5,33,25,35,19]. In the multi-domain frameworks, the low-level recognition is usually performed independently from the context, through the identification of graphical primitives (lines, arcs, ellipses, etc.). In the most advanced products, the domain knowledge is then used at a higher level, to correct possible low-level interpretation errors.

The instantiation of a framework for a specific domain often requires the definition of a visual grammar, which can only be performed by particularly skilled programmers. In fact, for some languages (such as in [23,11]) the grammar is composed of several tens of productions.

In this paper, we present a new methodology aimed at developing a framework for the recognition of sketched diagrams from different domains. The main innovation is the introduction of a recognition phase based on the analysis of the local context of symbols. This is effective since many visual languages need to be simple in order to

☆ This paper has been recommended for acceptance by Shi Kho Chang.
* Corresponding author.
E-mail addresses: gencos@unisa.it (G. Costagliola), matderosa@unisa.it (M.D. Rosa), vfuccella@unisa.it (V. Fuccella).

http://dx.doi.org/10.1016/j.jvlc.2014.10.021
1045-926X/

be used, and as a result their structure happens to be simple enough to be captured with local checks. We demonstrate this by showing that binary trees and even a fairly complex flowchart dialect can be fully syntactically modelled through this approach. From the point of view of sketch recognition, one of the innovations introduced with this framework is that it learns directly from sample sketches of the specific domain the information used for low-level recognition, taking advantage of the various innovative machine learning-based techniques produced in recent years (see next section).

The rest of the paper is organised as follows: the next section contains a brief survey of related work, including visual language specification and recognition, with details on sketch recognition; Section 3 describes our methodology for the definition of a language by using the local context of each symbol; in Section 4 we describe the framework, its architecture, the main recognition techniques, and the data provided in order to instantiate the framework for a particular domain; lastly, some final remarks and a brief discussion on future work conclude the paper.

2. Related work

In this brief survey, we focus on both visual languages for diagram recognition and their applications exploiting a sketch-based interaction paradigm, with emphasis on multi-domain frameworks.

2.1. Visual language specification

In recent years many methods to model a diagram as a sentence of a visual language have been devised. A diagram has been represented either as a set of relations on symbols (the *relation-based approach*) [36] or as a set of attributed symbols with typed attributes representing the "position" of the symbol in the sentence (the *attribute-based approach*) [22]. Even though the two approaches appear to be different, they both model a diagram (or visual sentence) as a set of symbols and relations among them. Unlike the relation-based approach, where the relations are explicitly represented, in the attribute-based approach the relations must be derived by associating attribute values. The former approach is therefore at a higher level of abstraction with respect to the latter.

Based on these models, several formalisms to describe visual language syntax with scanning and parsing techniques have been proposed: Relational Grammars [40], Constrained Set Grammars [31], and (Extended) Positional Grammars [15]. In [29], these are reported as non-graph grammars, as opposed to graph grammars. In general, non-graph grammars are specified by providing an alphabet of graphical symbols with their "physical" appearance, a set of spatial relationships generally defined on symbol position and attachment points/areas, and a set of grammar rules in context-free like format. Although the grammars are non-graph, the spatial relationships can be represented by using the so-called spatial-relationship graph [4] in which each node corresponds to a graphical symbol and each edge corresponds to the spatial relationship between the symbols.

A large number of tools are able to create WIMP development environments for visual languages. These are usually based on grammars and include, among others, VLDesk [12], DiaGen [32], GenGed [3], Penguin [6], AToM3 [17], and VL-Eli [27]. However, to the best of our knowledge, no tools for multi-domain sketch development environments for visual languages have been devised so far.

Despite the context-free like rule format, the grammars for visual languages are not easy to define and read. This may explain why there has not been much success in transferring these techniques from research labs into real-world applications. We observe that several visual languages in use today are simple languages that focus on basic components and their expressive power. These languages do not need to be described with complex grammar rules. Our approach provides a simpler specification for many of them, making it easier to define sketched languages and quickly prototype visual languages.

2.2. Multi-domain sketch recognition

In the literature of sketch recognition we can find the description of many solutions, both multi-domain and domain-specific (e.g., UML diagrams [3], electrical circuits [4], chemical drawings [5], etc.). Most multi-domain approaches [1,5] exploit domain knowledge to improve recognition at a lower level. *SketchREAD* [1] performs recognition by using a structural description of the domain symbols. In *AgentSketch* [5] an agent-based system is used for interpreting sketched symbols. *LADDER* is a language [25] which can automatically generate a domain specific sketch recognition system. It enables the definition of sketched elements at different levels (e.g., primitives, symbols, entire diagrams, etc.). *InkKit* [7] works very similarly to our framework: it firstly classifies the strokes as either writing or drawing, then identifies the basic shapes such as lines, rectangles, and circles, and the relationships between them. Compared to our approach, most of the multi-domain frameworks do not allow a simple and/or complete definition of how to interpret high-level diagrams.

Some frameworks aim to solve specific subproblems of sketch recognition, e.g.: stroke fragmentation [42,26] and recognition of unistroke [37,41] and multi-stroke [30,21,33] symbols. The objective of stroke fragmentation is to recognize the graphical primitives composing the strokes and is often used in structural symbol recognition.

As for symbol recognition, a widely accepted taxonomy [44] classifies the methods in two main categories: *structural* and *statistical*. In *structural* methods [30,10,16], symbol matching is performed by finding a correspondence between structures, such as graphs or trees. In *Statistical* methods [21,33], symbol features are extracted and compared. Advances in unistroke symbol recognition make it useful for gesture recognition [41,14] in various applications, including interfaces for mobile devices [43,20].

3. Defining visual languages using local context

We propose a new methodology for the recognition of sketched diagrams. The key to our approach is a new local