Context-Driven Constraint Multiset Grammars with Incremental Parsing for On-line Structured Document Interpretation

Sébastien Macé Eric Anquetil

IRISA - INSA
Campus Universitaire de Beaulieu
35042, Rennes Cedex, France
{Sebastien.Mace, Eric.Anquetil}@irisa.fr

Abstract

This paper deals with the incremental interpretation of on-line structured documents. We introduce Context-Driven Constraint Multiset Grammars (CDCMG) *to model the analysis of strokes drawn in the context of document composition; they allow visual arrangement, constraint and stroke recognition system handling by coupling a global vision of the document with a local vision of the analyzed elements. The associated incremental parser interprets the hand-drawn strokes and allows immediate feedback to the user. Moreover, we extend CDCMG by exploiting fuzzy logic to take into account the imprecision of handwriting and explain the impact on ambiguity management. Thanks to this approach, pen-based interfaces have been designed; we present and evaluate more specifically a software for electric diagram drawing, both in terms of comparison with more usual button-based interfaces and recognition rates.*

1. Introduction

On-line hand-drawn stroke recognition is a complex pattern recognition problem due to the variability of handwriting styles. After many years of research, it is now possible to design very powerful recognition systems for *isolated hand-drawn shapes*, but these ones are not sufficient in the context of *on-line structured documents* such as mathematical formulas, musical scores or electric diagrams. This is because such documents are composed of many symbols of various natures. Moreover, a stroke can have various different interpretations depending on its context (structural and/or temporal) which must therefore be taken into account. As a consequence, it remains a challenge to develop software taking advantage of the intuitiveness of pen interaction to draw complex on-line structured documents on tablet PCs. It is even more problematic for *interactive software design*, in which the interpretation process must be incremental, i.e. analyze the strokes and replace them by their corresponding symbols during the drawing of the document. This allows immediate interactive error correction and implicit validation by the user. In this paper, we investigate *generic methods* for such purposes.

Although there are many systems dedicated to one specific nature of documents (e.g. mathematical formulas, musical scores, etc.), there are few generic approaches modeling the incremental recognition of on-line structured documents. They are mainly based on the segmentation of the strokes into primitives, such as lines or arcs, and the combination of these primitives according to various kinds of rules which can model many natures of documents [1][5][9]. This may offer the advantage of less constraining the user to one specific way of drawing the document (e.g. one stroke can be segmented into a sequence of several primitives), it nevertheless has three major drawbacks. Firstly, this *structural* approach does not take advantage of *morphological* hand-drawn shape recognition systems (based for instance on hidden Markov models or support vector machines). Secondly, the recognition is completely dependent on this segmentation process because existing approaches mostly do not reconsider segmentation whatever is the result of the following steps. Thirdly, we believe that it is not always possible to describe all complex symbols as a combination of such basic primitives. Thus, this segmentation process is probably not always suitable.

Moreover, existing rules used to combine primitives are often context-free: they define a local arrangement of these primitives and do not have a global vision of the document (i.e. do not take into account other document elements), which is not sufficient for complex hand-drawn document interpretation. Marriott introduced *Constraint Multiset Grammars* (*CMG*) [8] and their associated incremental parser which offer the advantage of being context-sensitive. In [5], Chok and Marriott say that it can be exploited in the context of hand-drawn diagrams; nevertheless, the pattern recognition and segmentation problematics are not discussed. We previously [6] proposed a generic approach based on context-sensitive interpretation rules modeling the coupling of a global vision of the analyzed document with a local vision of the shape to recognize, and making it possible to exploit morphological hand-drawn shape recognition systems. It has been used to design several pen-based interfaces, for instance for the drawing of musical scores and UML diagrams [6]. This paper aims at defining a well formalized merging of CMG with our previous works.

This paper is organized as follows. Next section introduces the grammatical formalism and its associated incremental parser. Section 3 focuses on the fuzzification of the formalism and the consequence on ambiguity management. Then, we present and evaluate a system for electric diagram editing based on this method. Finally, we draw some conclusions and suggest directions for future work.

2. Context-Driven Constraint Multiset Grammars

Interpreting hand-drawn structured documents requires methods for bi-dimensional syntactical analysis: it is necessary, on the one hand, to identify the strokes drawn by the user, and, on the other, to establish the global structure of the document. It is our opinion that the second should drive the first to facilitate the analysis. In this section, we propose a new grammatical formalism for such purpose.

2.1. Formal Description

Constraint Multiset Grammars (*CMG*) are an instance of *Attributed Multiset Grammars*, which means they associate attributes to the elements of the visual alphabets they handle. A CMG G is a quadruple $G = (V_N, V_T, P, S)$ where V_N, V_T, P are sets of *non-terminals*, *terminals* and *productions*, respectively, and S is the *starting symbol*. A production rule in P can have the form

$$\alpha \rightarrow \beta \; where \; \{C\} \; and \; (D)$$

where α and β are multisets in $(V_T \cup V_N)^+$, C are constraints to satisfy and D are definition of the attributes of the elements in α. CMG allow the use of both existential and negative constraints in C, which makes them context-sensitive. As $\alpha \in (V_T \cup V_N)^+$, productions are *unrestricted*, i.e. more than one symbol can be produced. Concretely, this makes segmentation possible.

As introduced previously, CMG as used in [5][8] are not sufficient to deal with complex hand-drawn structured documents. Like most authors, we define domain-independent primitives, but instead of segmenting the strokes into smaller predefined primitives, we use only one primitive: the hand-drawn stroke. It means that $V_T = \{stroke\}$, whereas V_N is domain-dependent and is the set of all the possible visual symbols of the document. Then, we adapt the formalism to define *Context-Driven Constraint Multiset Grammars* (*CDCMG*), in which C is dedicated to parsing driving. Productions are formalized as follows [6]:

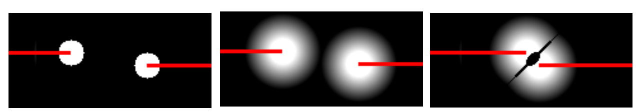
$\alpha \rightarrow \beta \; where \; \{$
1. $document \; context \; verification \; (DCV) \; block.$
2. $constraint \; verification \; (CV) \; block.$
3. $shape \; recognition \; (SR) \; block.$
4. $document \; context \; creation \; (DCC) \; block.$

$\} \; and \; (D)$

These blocks of constraints have a predefined organization. They model the coupling of a global vision of the document (DCV and DCC blocks) with a local vision of β (CV and SR blocks). This block organization allows both natural constraint encoding and efficient parsing.

DCV and DCC blocks: These two blocks aim, on the one hand, at modeling a global vision of the document physical structure, i.e. expressing how its elements are relatively positioned, and, on the other hand, at efficiently driving the analysis process. DCV and DCC blocks are based on the definition of *document structural contexts*, which are specific locations in the document in which some particular elements can exist; they can be considered as structural constraints. The DCV block is a list of constraint preconditions that model the structural contexts in which the elements of β have to be located for the production to be tested, whereas the DCC block is a list of postconditions that model which structural contexts have to be created due to the reduction of the production, i.e. which new elements can now exist and where.

Figure 1(a) presents two document structural contexts modeling how, in a diagram drawing context, a forthcoming vertical segment must be positioned to be connected to one of the two already existing horizontal segments. Such contexts can be denoted as $\{HSeg[extremity]VSeg[any]\}$: to satisfy the context, at the positioning $extremity$ relatively to a $HSeg$, a $VSeg$ must have *any* of its points.



(a) Crisp contexts (b) Fuzzy contexts (c) Using reject

Figure 1. Two structural contexts (high grey values model high membership values).

CV and SR blocks: These two blocks model a local vision of the elements in β. The CV block explicits local conditions and allows natural modeling of both constraints and relative positioning. In order to strengthen this structural recognition, morphological hand-drawn shape recognition systems can be exploited; it is modeled by the SR block which allows using recognizers that are dedicated to family symbols and that return class scores. We have stated in [6] that the coupling of these two blocks defines a canonical on-line signal form: the constraints in the CV block identify each element in β and allow to call the recognition systems in the SR block with always the same entry order.

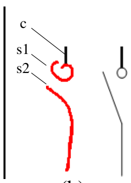
2.2. Examples of productions

Figure 2(a) presents a production for the interpretation of the *Switch* electric component. It models that a switch is drawn with two strokes $s1$ and $s2$ located below a same already existing connection c (DCV block). The stroke $s2$ must be below and near the stroke $s1$ (CV block). When sent to the component recognition system (in a predefined order), this one must recognize $s1$ and $s2$ as a switch (SR block). Finally, the existence of a switch allows drawing one connection below it (DCC block). Please note that there is no constraint on the drawing order of the strokes $s1$ and $s2$.

$Switch : res \rightarrow Stroke : s1\ ,\ Stroke : s2\ where\ \{$
1. $DCV:\ \exists\, Connection : c\ where$
   $\{c[below]s1[all]\}\ \&$
   $\{c[below]s2[all]\}$
2. $CV:\quad belowNear(s1, s2)$
3. $SR:\quad componentReco(\{s1, s2\}, Switch)$
4. $DCC:\ \{res[below]Segment^1[any]\}$

$\}\ and\ (res.connection = c).$

(a)



(b)

Figure 2. Interpretation of a switch.

On the contrary of other existing formalisms, CDCMG do not rely on segmentation of the strokes into primitives (cf. figure 2(b): the segmentation of the stroke $s2$ into two segments may be difficult and unstable). Nevertheless, it is also possible to model that a switch can be drawn with three strokes thanks to another production.

Please note that $connection$ is a particular attribute of the $Switch$ visual symbol. Moreover, some aliases are used to specify, for instance, that a $VSegment$ is a particular $Segment$.

2.3. Analysis Process

Interactive on-line structured document analysis requires efficient real-time incremental parsing to be usable. For that purpose, we have designed an *incremental parser* associated to CDCMG. It performs in a bottom-up strategy, and consists in trying to repeatedly reduce multisets of document elements into other multisets. The general aspects of incremental parsing of CMG are presented in [8]. The main originality is the *deterministic* aspect of the parser. It is an interesting property for interactive system design: thanks to the visual feedback, the user is aware of the interpretations of the system and then can be integrated in the analysis process, by implicitly validating or explicitly rejecting its answers. As a consequence, the parser should not reconsider a production that has been reduced previously because it has been implicitly considered as valid by the user.

The main analysis process modification we propose is due to the introduction of the two new DCV and DCC blocks. Indeed, document structural contexts act as triggers: when looking for productions to apply on an analyzed element, the principle consists in determining in which of these contexts (created in some DCC blocks) it is located and activating the corresponding productions. Then, for a production to be tested, all its DCV contexts must be satisfied. This gives predictive capabilities to the parser, which allows a substantial gain of time, because only structurally consistent productions are tested. This context-driven analysis process also implies that only context-relevant stroke recognition systems are used by the parser.

3. Context-Driven Fuzzy Constraint Multiset Grammars

The challenge of this section is to take the imprecision of handwriting into account. We propose a new extension of CDCMG by exploiting fuzzy logic, which is well adapted to manage such imprecise data and to avoid making a crisp decision. *Context-Driven Fuzzy Constraint Multiset Grammars* (*CDFCMG*) are a *fuzzification* of CDCMG: their definition is the same, except that productions are now of the form

$$\alpha \xrightarrow{\rho} \beta\ where\ \{C\}\ and\ (D)$$

with $\rho \in [0, 1]$ modeling the reliability value for β to be reduced into α, i.e. the degree to which a production is satisfied. It is necessary to define ρ_P for each production P. In this work, ρ_P is context-dependent, i.e. it is not set a priori.

3.1. Defining ρ_P for a production P

ρ_P is defined as a fuzzy combination (a t-norm) of the constraints in the blocks of C as follows (μ_P^X is the reliability value of the block X of the production P):

$$\rho_P = \text{t-norm}(\mu_P^{DCV}, \mu_P^{CV}, \mu_P^{SR}) \qquad (1)$$

Thus, the verification of each block must be modeled qualitatively. We focus on the computing of μ_P^{DCV}, which is the most original part of the formalism; indeed, defining fuzzy constraints in the CV block is often straightforward and recognition systems used in the SR block are based on fuzzy inference systems from which we obtain such degrees [2].

Defining a fuzzy landscape μ_{pos}: As introduced previously, a DCV block is defined as a list of document structural contexts which can be denoted $\{R[position_R]A[part_A]\}$, which means that the analyzed element A must have its part $part_A$ at the relative positioning $position_R$ of the reference R. The presented method is inspired by the works of Bloch [3]. We a priori associate a fuzzy landscape μ_{pos} to each position $position_R$; it models for each point of the document its membership to the relative position. Such function can be defined empirically or learned from examples [4]. Figure 1(b) presents the result of the fuzzification of the structural contexts presented previously.

Computing $\mu_{pos}(A)$: The next step consists in determining the degree $\mu_{pos}(A)$ to which A is in the fuzzy landscape. It depends on the value of $part_A$; some examples are presented in table 1.

$part_A$	$\mu_{pos}(A)$
one specific point x ($\{R[\_]A[first\_point]\}$, $\{R[\_]A[highest\_point]\}$, etc.)	$\mu_{pos}(x)$
any point ($\{R[\_]A[any]\}$)	$\max_{x \in A} \mu_{pos}(x)$
all the points ($\{R[\_]A[all]\}$)	$\frac{1}{\lvert A \rvert} \sum_{x \in A} \mu_{pos}(x)$

Table 1. Definition of $\mu_{pos}(A)$ depending on the value of $part_A$.

Deducing μ_P^{DCV}: Finally, μ_P^{DCV} is defined as a normalized t-norm of all the $\mu_{pos}(A)$ in the block. The normalization aims at not giving an advantage to productions with few contexts to satisfy.

3.2. Impact on the analysis process

The analysis process is globally not changed by the fuzzification of the grammatical formalism, except for decision making, in which we now exploit reject options. This time, a production P is applicable if ρ_P is high enough, which models an *outlier reject*. When several productions are applicable, we select the best satisfied, i.e. the one with the highest ρ.

Nevertheless, several productions can have very close values of ρ, which implies an ambiguity between several interpretations: it may then not be pertinent to choose one in particular but rather to reject all of them (and to let the user re-draw the stroke). For that purpose, we exploit the *ambiguity reject* [10] and compare the scores ρ_{P_1} and ρ_{P_2} of the two best productions P_1 and P_2, such that $\rho_{P_1} \geq \rho_{P_2} > 0$. Then we define a *reliability function* ψ_{P_1,P_2} as:

$$\psi_{P_1,P_2} = \frac{\rho_{P_1} - \rho_{P_2}}{\rho_{P_1}} \quad (2)$$

Finally, we use a threshold for ψ_{P_1,P_2} above which it is not pertinent to make a decision.

Figure 1(c) illustrates the consequence of using reject options on the example introduced previously when the two segment extremities are close: a reject area appears where the two corresponding fuzzy landscapes have close values.

4. Experimental results

This section presents the evaluation of a pen-based electric diagram prototype developed using the presented methodology. This interface allows to draw electric components from 22 different classes; it required the definition of about 35 productions. The stroke recognition systems used were based on fuzzy inference systems [2]. Figure 3(a) presents a screenshot of an on-line electric diagram drawn on a tablet PC, and figure 3(b) displays the corresponding recognized document. Although it is not visible on the figure, the stroke analysis process is incremental. It is also important to note that we use *graphical completion*: when users draw the first stroke of a multi-stroke symbol, if it is enough to deduce the complete symbol without any ambiguity, then the system directly displays it; this allows a substantial gain of time.

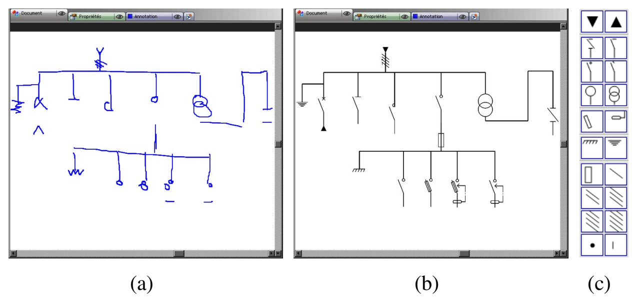


Figure 3. The pen-based electric diagram editor based on CDFCMG.

The evaluation of the system aims at qualifying the contribution of pen-based interfaces over more usual ones. For that purpose, we have developed the complementary button-based prototype in which users have to select electric components in a button bar (cf. figure 3(c)). Only connexions can be drawn in the same way as in the pen-based system.

Nine writers took part in the experiment performed on a tablet PC; three of them were used to both of the systems, whereas three others had never used a tablet PC. Writers were asked twice to draw four predefined electric diagrams with both of the prototypes, beginning alternatively by one or the other. Writers were asked to take all the time they needed to get used to the systems before taking part in the experiment; this step always took less than twenty minutes, which is a sign of ease of getting used to of the interface.

The results of this experiment, presented in table 2, emphasize a significant contribution from an ergonomic point of view: when drawing the same diagram with both of the interfaces, the pen-based system enables users to save, on average, about 32.5% time. The minimum gain was of 19.5%, whereas two of the people used to the systems saved more than 50% time. Indeed, when using a pen, users can directly draw the symbols where they want. This implies that they only have two *focuses of attention* (the diagram to reproduce and the one in the process of construction) against three in the button-based one (the diagrams plus the bar), which increases its user-friendliness.

We then focused more specifically on the pen-based prototype in order to evaluate its performances. For that purpose, we analyzed the error rates (when a stroke is misunderstood by the system and has to be deleted and re-drawn by the user) and reject rates (when a stroke is deleted by the system and only has to be re-drawn) during its using in the last experiment. The results, based on a total of 3956 strokes from 22 classes of symbols, are presented in table 2. The average error rate is 2.4%; therefore, the prototype is usable. The average reject rate is 9.3%, but it is less constraining for the user because he only has to re-draw the corresponding symbol. Actually, we give advantage to the reject in order to limit the errors, which is better from an ergonomic point of view and less intrudes in the user creative process.

writer	gain of time (%)	reject rate (%)	error rate (%)
# 1	29.4	14.1	1.5
# 2	53.1	3.0	1.0
# 3	37.4	3.5	1.2
# 4	37.2	10.6	2.6
# 5	22.4	12.2	4.8
# 6	50.9	10.2	1.8
# 7	19.5	9.8	2.7
# 8	23.6	9.6	2.8
# 9	19.5	8.8	1.2
average	32.5	9.3	2.4

Table 2. Gain of time, reject and error rates.

Actually, the interface proposes some other features, such as pen-based copy and paste. Moreover, it is possible to replace an electric component by another by drawing the new one above the old one; this makes recognition errors less constraining. This possibility is encoded thanks to CDCMG. Please note that the use of these two features was not allowed for the tests in order not to give an advantage to the pen-based prototype. Interested readers can refer to [7] for more details and some other evaluations.

5. Conclusion

In this paper, we have introduced *Context-Driven Constraint Multiset Grammars*, which are a merging of two existing approaches [5][6]. It offers the advantage of coupling a global vision of the document with a local vision of the analyzed elements, which allows both incremental document interpretation modeling and efficient parsing. The CDFCMG, fuzzification of this formalism, help dealing with imprecise hand-drawn strokes more easily. The evaluation of a pen-based electric diagram editor developed thanks to the presented methodology emphasizes the interest of such systems in comparison with more usual ones.

Our future research will focus on improving the decision making process of the CDFCMG parser; so far, we only evaluate grammatical rules one by one, and do not take rule sequences into account. Thus, our goal will be to evaluate the reliability value of the compound results. Moreover, we will estimate the complexity of the system depending on the number of symbols it proposes.

6. Acknowledgments

The authors would like to thank Guy Lorette, Professor at the University of Rennes 1, and all the people who took part in the experiments for their contribution to this work. This project benefits from the financial support of the Brittany Region.

References

[1] C. Alvarado and R. Davis. Dynamically constructed bayes nets for multi-domain sketch understanding. In *Proc. of IJCAI'05*, pages 1407–1412, 2005.

[2] E. Anquetil and H. Bouchereau. Integration of an on-line handwriting recognition system in a smart phone device. In *Proc. of ICPR'02*, pages 192–195, 2002.

[3] I. Bloch. Fuzzy relative position between objects in image processing: a morphological approach. *IEEE Transactions on PAMI*, 21(7):657–664, 1999.

[4] F. Bouteruche, S. Macé, and E. Anquetil. Fuzzy relative positioning for on-line handwritten stroke analysis. In *Proc. of IWFHR'06*, 2006.

[5] S. S. Chok and K. Marriott. Automatic generation of intelligent diagram editors. *ACM Transactions on Computer-Human Interaction*, 10(3):244–276, September 2003.

[6] S. Macé and E. Anquetil. A generic approach for eager interpretation of on-line handwritten structured document. In *Proc. of ICPR'06*, pages 1106–1109, 2006.

[7] S. Macé and E. Anquetil. Design of a pen-based electric diagram editor based on context-driven constraint multiset grammars. In *Proc. of HCII'07, LNCS 4551*, pages 418–428. Springer-Verlag, 2007.

[8] K. Marriott. Constrain multiset grammars. In *Proc. of VL'94*, pages 118–125, 1994.

[9] J. Mas, G. Sanchez, and J. Llados. An adjacency grammar to recognize symbols and gestures in a digital pen framework. In *Proc. of IbPRIA'05, LNCS 3523*, pages 115–122. Springer-Verlag, 2005.

[10] H. Mouchère and E. Anquetil. A unified strategy to deal with different natures of reject. In *Proc. of ICPR'06*, pages 792–795, 2006.