

A hybrid (visual/natural) controlled language

Yannis Haralambous¹  · Julie Sauvage-Vincent¹ ·
John Puentes¹

Published online: 22 February 2016
© Springer Science+Business Media Dordrecht 2016

Abstract We define the notion of *controlled hybrid language* that allows information share and interaction between a controlled natural language (specified by a context-free grammar) and a controlled visual language (specified by a Symbol-Relation grammar). We present the controlled hybrid language INAUT, used to represent nautical charts of the French Naval and Hydrographic Service (SHOM) and their companion texts (*Instructions nautiques*).

Keywords Controlled hybrid language · Controlled visual language · Controlled natural language · Electronic navigational charts · Maritime navigation

1 Introduction

In his milestone survey paper (Kuhn 2014), Kuhn excludes *visual languages* from his classification of controlled natural languages:

As a further remark, we should note that the term language is used in a sense that is restricted to sequential languages and excludes visual languages such as diagrams and the like.

Nevertheless visual languages can be formally defined controlled languages differing from controlled natural ones only in terms of modality. In the last 50 years a large number of visual languages have been defined and—similarly to

✉ Yannis Haralambous
yannis.haralambous@telecom-bretagne.eu

¹ Télécom Bretagne & UMR CNRS 6285 Lab-STICC, Institut Mines-Télécom, Technopôle Brest Iroise CS 83818, 29238 Brest Cedex 3, France

controlled natural languages—their specifications have used a large spectrum of algebraic, logical or grammatical methods (cf. Marriott et al. 1998).

Sometimes a controlled language can have more than one modality, like in the case of a controlled natural language defined in 2014 by Camilleri et al. (2014) which is in fact a textual modality for a controlled visual language of contract-oriented diagrams. Also in Kerpedjiev (1992) describes a system which produces simultaneously a text (a weather report for Bulgaria) and a weather map containing some of the information of the text.

We go one step further and consider a controlled language that is simultaneously the textual *and* the visual representation of a knowledge base in the first-order logic formalism. We argue that text and image should not only coexist but also be complementary and that the two modalities should interact. When “writing” in this type of language, the author shares information between the textual and the visual modality, depending on the context and her needs. We call such a language, a *controlled hybrid language*.

In this paper we present a typical case of controlled hybrid language: geographic maps and their companion texts. As Varanka states in (1991, p. 285):

Any document consisting of both maps and accompanying geographical descriptions or explanatory texts employ[s] the characteristics of both. [...] The premise is that texts are decisive to any reading of the maps and that they lend substance to the abstract and highly codified cartographic images. Without the text the map gives a minimum of information but with it they take on new meanings that are crucial to understanding the geographical attitudes and thoughts held by their users. Thus the arrangements of maps with texts have a critical, yet subtle effect on the nature of geographical information.

This was written in 1991, when Electronic Navigational Charts (ENCs) were neither as widespread, nor as powerful as today. An ENC is a structured electronic document containing geolocated objects with clearly defined semantics.

A map can be represented by an ENC, but *is it a language?* This has been a long debate (see Grant Head 1991; Li 1995) and many theoreticians are inclined to answer positively. Instead of pursuing the debate for general geographic maps and companion texts, we will present a specific case where maps are considered as language: a controlled hybrid language for the nautical charts of the French Naval and Hydrographic Service (SHOM) and their companion texts (*Instructions nautiques*).

The controlled hybrid language INAUT (cf. Sect. 5) uses a standard context-free grammar for the textual modality and a Symbol-Relation grammar (cf. Sect. 2.1) for the visual modality. A hybrid sentence uses both modalities and shares the information among them in various degrees: for example, a given hybrid sentence can be represented in text-only, or by a map and accompanying text, or (in some cases) by a map-only. Furthermore, text and map can interact using the GUI of ENC visualization devices.

The structure of this paper is as follows: after this general introduction, we introduce the reader to controlled visual languages (Sect. 2) and then to controlled

hybrid languages (Sect. 3). In Sect. 4 we discuss how maps with companion text can be represented by controlled hybrid languages, and in Sect. 5 we describe the specific case of the INAUT hybrid language. The rest of the paper is dedicated to the description of tools used (Sect. 6), to the evaluation (Sect. 7) and to the conclusion and future plans (Sect. 8).

This paper is an extended and revised version of Haralambous et al. (2014), presented at CNL 2014 in Galway, Ireland.

2 Controlled visual languages

A visual language is formally defined by a Symbol-Relation (SR) grammar (Ferrucci et al. 1996). In the following sections we will first describe SR grammars and then visual languages based on SR grammars.

2.1 From standard context-free grammars to SR grammars

Ever since Chomsky's seminal 1956 paper (Chomsky 1956), formal grammars as a means to define formal languages have become a classical computer science tool. Let us take a simple example of a Chomsky Type 2 (context-free) grammar: consider the alphabet $\{a, b\}$, the set of nonterminals $\{A, S\}$ with start symbol S , and the rewrite rules

$$s_0: S \rightarrow A \quad (1)$$

$$s_1: A \rightarrow aAb \quad (2)$$

$$s_2: A \rightarrow ab. \quad (3)$$

This context-free grammar defines the language of words $a^n b^n$ for $n \geq 1$. To obtain, for example, word $aabb$ one can apply following derivations:

$$S \xrightarrow{s_0(S)} A \xrightarrow{s_1(A)} aAb \xrightarrow{s_2(A)} aabb. \quad (4)$$

Let us now represent the word $aabb$ in a different way: *let us index the occurrences of the same symbol by superscripts and let us consider the property of being adjacent in the word as a binary relation which we will call "next"*. Then, the word $aabb$ can be written as a pair of sets¹:

$$\langle \{a^1, a^2, b^2, b^1\}, \{\text{next}(a^1, a^2), \text{next}(a^2, b^2), \text{next}(b^2, b^1)\} \rangle. \quad (5)$$

¹ Since $\{a^1, a^2, b^2, b^1\}$ is a set, there is *no* intrinsic order among its elements. But there is nevertheless an order in the word $aabb$ and, despite appearances, this order is *not* provided by the numeric superscripts (which can be arbitrary), but *solely* by the "next" relations of set $\{\text{next}(a^1, a^2), \text{next}(a^2, b^2), \text{next}(b^2, b^1)\}$.

In the remainder of this paper we will use the following notation: in derivations, nonterminals will be denoted by letters with superscripts (whenever new symbols are written, superscripts will be incremented); in rewrite rules, nonterminals (except for the start symbol S) will be denoted by symbols $\circ, \square, \triangle, \nabla$, etc., where \circ will be systematically placed in the left part of the rule. This notation (which deviates from the original Ferrucci notation, Ferrucci et al. 1996) will make it easier to follow the use of rewrite rules in derivations.

Using this convention and the notation of (5), the three rewrite rules become:

$$s_0: S \rightarrow \langle \{\square\}, \emptyset \rangle \tag{6}$$

$$s_1: \circ \rightarrow \langle \{a^1, \square, b^1\}, \{\text{next}(a^1, \square), \text{next}(\square, b^1)\} \rangle \tag{7}$$

$$s_2: \circ \rightarrow \langle \{a^1, b^1\}, \{\text{next}(a^1, b^1)\} \rangle . \tag{8}$$

Let us now write the first two derivations:

$$S \xrightarrow{s_0(S)} \langle \{A^1\}, \emptyset \rangle \tag{9}$$

$$\xrightarrow{s_1(A^1)} \langle \{a^1, A^2, b^1\}, \{\text{next}(a^1, A^1), \text{next}(A^1, b^1)\} \rangle . \tag{10}$$

In (9), \square has been written A^1 , and in (10), A^1 is rewritten by a new symbol, we have called this symbol A^2 . But there is a problem in (10): in the “next” relations we still find symbol A^1 , despite the fact that it has been rewritten by rule s_1 in the set of symbols. To correct this inconsistency we introduce a new kind of rules r_* , which are applied to relations immediately after s_* rules:

$$r_{\{1\},1}: \text{next}(*, \circ) \rightarrow \{\text{next}(*, \square)\} \tag{11}$$

$$r_{\{1\},2}: \text{next}(\circ, *) \rightarrow \{\text{next}(\square, *)\}, \tag{12}$$

where $*$ is any terminal or nonterminal, and \circ, \square represent the same symbols as in s_1 . In the notation $r_{\{1\},1}$, the first index is the set of s_* rules after which we are allowed to use this rule, and the second index identifies this rule among those having the same set of compatible s_* rules. Let us continue our derivations from (10) using r_* rules:

$$(10) \xrightarrow{r_{\{1\},1}(\text{next}(a^1, A^1))} \langle \{a^1, A^2, b^1\}, \{\text{next}(a^1, A^2), \text{next}(A^1, b^1)\} \rangle \tag{13}$$

$$\xrightarrow{r_{\{1\},2}(\text{next}(A^1, b^1))} \langle \{a^1, A^2, b^1\}, \{\text{next}(a^1, A^2), \text{next}(A^2, b^1)\} \rangle , \tag{14}$$

where in both (13) and (14), \circ becomes A^1 and \square becomes A^2 .

Now it is time to apply s_2 :

$$(14) \xrightarrow{s_2(A^2)} \langle \{a^1, a^2, b^2, b^1\}, \{\text{next}(a^1, A^2), \text{next}(a^2, b^2), \text{next}(A^2, b^1)\} \rangle \tag{15}$$

and we need new r_* rules to get rid of A^2 in the relations:

$$r_{\{2\},1}: \text{next}(*, \bigcirc) \rightarrow \{\text{next}(*, a^2)\} \tag{16}$$

$$r_{\{2\},2}: \text{next}(\bigcirc, *) \rightarrow \{\text{next}(b^2, *)\}, \tag{17}$$

which give the following derivations:

$$(15) \xrightarrow{r_{\{2\},1}(\text{next}(a^1, A^2))} < \{a^1, a^2, b^2, b^1\}, \{\text{next}(a^1, a^2), \text{next}(a^2, b^2), \text{next}(A^1, b^1)\} > \tag{18}$$

$$\xrightarrow{r_{\{2\},2}(\text{next}(A^2, b^1))} < \{a^1, a^2, b^2, b^1\}, \{\text{next}(a^1, a^2), \text{next}(a^2, b^2), \text{next}(b^2, b^1)\} > , \tag{19}$$

and we are done, since (19) is exactly (5).

The user may argue that (6)–(19) are just a more complex way to obtain the same result as (1)–(4) and this is true for this particular case. But amazing new perspectives arise when we go beyond the unique relation “next” and allow an arbitrary number of relations.

Let us call symbols, *s-items*, and relations, *r-items*. Rules such as s_0, s_1 and s_2 are called *s-productions*. S-productions have an s-item on the left side, and on the right side a pair of sets consisting of a set of s-items and a set of r-items. Rules such as r_1 and r_2 are called *r-productions*. R-productions have an r-item on the left side, and a set of r-items on the right side. A derivation step consists of applying an s-production s_i followed by zero, one or more r-productions $r_{\{i\},*}$ “compatible” with it. Every r-production must be “compatible” with one or more s-productions.

A grammar with s-items, r-items, s-productions and r-productions is called *Symbol-Relation grammar* (or shortened: SR grammar) (Ferrucci et al. 1996). More formally, an SR grammar is a sextuple $(N, T, R, S, \mathbf{s}, \mathbf{r})$ where N is a set of nonterminal symbols, T a set of terminal symbols, R a set of binary relations, S is the start symbol, \mathbf{s} a set of s-productions and \mathbf{r} a set of r-productions.

As in the case of standard formal grammars, we can define a grammar tree structure for SR grammars. It suffices to consider s-items and r-items as vertices and s-productions and r-productions as edges. In Fig. 1, for example, the reader can see the SR tree structure of the word $< \{a^1, a^2, b^2, b^1\}, \{\text{next}(a^1, a^2), \text{next}(a^2, b^2), \text{next}(b^2, b^1)\} > .$

Let us see now how to obtain the semantics of the word through the SR grammar tree structure. Following Knuth’s approach (1968), we attach attributes to the leaves of the syntax tree (s-items, r-items) and calculate the values of these attributes for non-leave nodes, in a bottom-up approach based on attribute-value rules corresponding to the rewrite rules of the grammar. For example, if we consider the semantics of terminal s-items to be integer numbers and concatenation to be multiplication, then the semantics of the word $aabb$ would be the product of the four values of terminal symbols. More precisely, let us consider $*_\sigma$ to be the numeric values of terminal symbols $*$, and let us write:

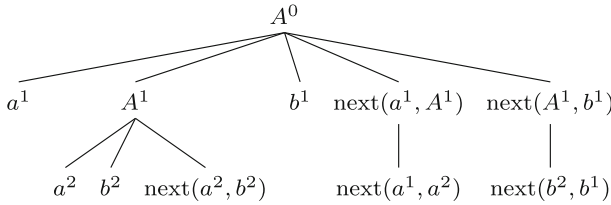


Fig. 1 The SR grammar tree structure of word $\langle \{a^1, a^2, b^2, b^1\}, \{next(a^1, a^2), next(a^2, b^2), next(b^2, b^1)\} \rangle$

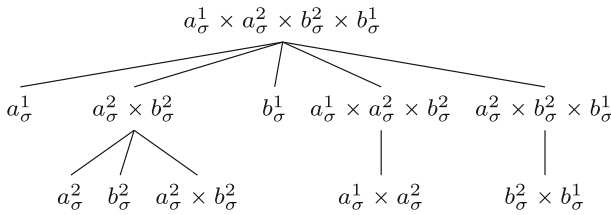


Fig. 2 The semantics of the SR grammar tree structure of word $\langle \{a^1, a^2, b^2, b^1\}, \{next(a^1, a^2), next(a^2, b^2), next(b^2, b^1)\} \rangle$

$$\begin{aligned}
 &next(*^1, *^2)_\sigma \leftarrow *^1_\sigma \times *^2_\sigma \quad (\text{where } * \text{ can be any terminal}) \\
 s_0: S^0_\sigma &\leftarrow \square_\sigma \\
 s_1: \bigcirc_\sigma &\leftarrow \frac{next(a^1, \square)_\sigma \times next(\square, b^1)_\sigma}{\square_\sigma} \\
 r_{\{1\},1}: next(*, \bigcirc)_\sigma &\leftarrow next(*, \square)_\sigma \quad (\text{where } * \text{ can be any symbol}) \\
 r_{\{1\},2}: next(\bigcirc, *)_\sigma &\leftarrow next(\square, *)_\sigma \quad (\text{where } * \text{ can be any symbol}) \\
 s_2: \bigcirc_\sigma &\leftarrow next(a^1, b^1)_\sigma \\
 r_{\{2\},1}: next(*, \bigcirc)_\sigma &\leftarrow next(*, a^2)_\sigma \quad (\text{where } * \text{ can be any symbol}) \\
 r_{\{2\},2}: next(\bigcirc, *)_\sigma &\leftarrow next(b^2, *)_\sigma \quad (\text{where } * \text{ can be any symbol}).
 \end{aligned}$$

In Fig. 2, the reader can see the values of nodes of the tree of Fig. 1 when the semantic rules above are applied, starting from the bottom (we consider multiplication \times to be commutative and associative). As can be seen in the figure, the value of the complete word is equal to the product of the values of the terminal symbols, as expected.

In the next section we will present an application of SR grammars to controlled visual languages.

2.2 An SR grammar based on topological logic

According to Pratt-Hartmann (2013), *topological logics* are “formal systems for representing and manipulating information about the topological relationships

between objects in space”. One of the first contributions in the field was the paper “A Spatial Logic based on Regions and Connection” by Randell et al. (1992), which introduced a logical theory (RCC8) for describing the relative position of regions, from a topological point of view. The mathematical frame is the one of topological spaces of which we take a set of nonempty regular closed subsets,² called *regions*. The logical theory is based on a single primitive binary predicate C . When interpreted in the domain of regions of a space, then a true $C(x, y)$ means that regions x and y share a common point.

Out of predicate C we define six new predicates: DC (disconnected), EC (external contact), PO (partial overlap), EQ (equality), TPP (tangential proper part), NTPP (nontangential proper part), giving the eight possible topological arrangements of two regions³ (cf. Fig. 3). Here is how these predicates are defined out of C (intermediate predicates P , O , PP have been included to make the formulas more intelligible) (Randell et al. 1992, p. 167):

$DC(x, y) := \neg C(x, y)$	(x and y are disconnected)
$P(x, y) := \forall z[C(z, x) \rightarrow C(z, y)]$	(x is a part of y)
$O(x, y) := \exists z[P(z, x) \wedge P(z, y)]$	(x and y have a common sub-region)
$EC(x, y) := C(x, y) \wedge \neg O(x, y)$	(x and y have a common point but not a common sub-region, i.e., they are externally tangential)
$PO(x, y) := O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$	(x and y have a common sub-region but none of them is sub-region of the other)
$EQ(x, y) := (x = y)$	(x and y are equal)
$PP(x, y) := P(x, y) \wedge \neg P(y, x)$	(x is a sub-region of y but y is not a sub-region of x)
$TPP(x, y) := PP(x, y) \wedge \exists z[EC(z, x) \wedge EC(z, y)]$	x is a proper part of y and there is a region z which is external tangential to both
$NTPP(x, y) := PP(x, y) \wedge \neg \exists z[EC(z, x) \wedge EC(z, y)]$	x is a proper part of y and there is no region z which is external tangential to both).

We define the SR-grammar SR-RCC8, based on the eight binary relations of RCC8. More precisely, the terminals of SR-RCC8 will be regions a^i ($i \geq 0$), nonterminals

² A regular closed set is the topological closure of the interior of some set.

³ Notice that these are *topological* and not *geometric* arrangements. In other words: the fact that we have used circles to draw shapes in Fig. 3 is unimportant, a region can be any shape, even non-connected and with “holes”. All that counts is whether x and y have, or not, a nonempty intersection, whether this intersection is a region or not, and whether x is, or not, included in y and vice-versa. For example, both in EC and in PO, the intersection of x and y is nonempty, the difference is that in PO the intersection $x \cap y$ is a region, while in EC, it is part of the boundary of x and of y , and hence is not a region.

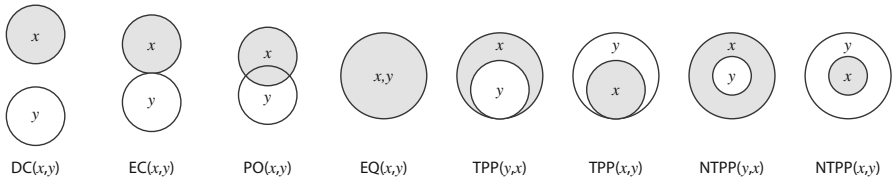


Fig. 3 The eight predicates of theory RCC8

will be written $\bigcirc, \square, \triangle, \nabla$ in the rules and A^i in the derivations, relations will be $\{DC, EC, PO, EQ, TPP, NTPP\}$ ⁴, the start symbol will be S . As for s-productions and r-productions, we will describe some of them through a couple of examples.

To start with, let us see how to describe by this grammar the visual sentence of Fig. 4 (numbers i in the figure designate regions a^i).

We start by rules that produce nonterminal TPP-related regions from the start symbol or from a nonterminal:

$$s_0: S \rightarrow \langle \{\square, \triangle\}, \{TPP(\triangle, \square)\} \rangle$$

$$s_1: \bigcirc \rightarrow \langle \{\square, \triangle\}, \{TPP(\triangle, \square)\} \rangle .$$

In the case of s_1 , we want \square to be the “reincarnation” of \bigcirc , in the sense that we want it to inherit all topological properties of \bigcirc . To attain this goal, we need r-productions

$$r_{\{1\},1}: @(\bigcirc, *) \rightarrow \{ @(\square, *) \}$$

$$r_{\{1\},2}: @(*, \bigcirc) \rightarrow \{ @(*, \square) \},$$

where $@ \in \{DC, EC, PO, EQ, TPP, NTPP\}$ and $*$ is any terminal or nonterminal symbol. We have the following derivations:

$$S \xrightarrow{s_0(S)} \langle \{A^1, A^2\}, \{TPP(A^2, A^1)\} \rangle \tag{20}$$

$$\xrightarrow{s_1(A^1)} \langle \{A^3, A^4, A^2\}, \{TPP(A^2, A^1), TPP(A^4, A^3)\} \rangle \tag{21}$$

$$\xrightarrow{r_{\{1\},2}(TPP(A^2, A^1))} \langle \{A^3, A^4, A^2\}, \{TPP(A^2, A^3), TPP(A^4, A^3)\} \rangle . \tag{22}$$

After renumbering superscripts, this sentence becomes

$$\langle \{A^1, A^2, A^3\}, \{TPP(A^3, A^1), TPP(A^2, A^1)\} \rangle . \tag{23}$$

It is clear that this sentence does not completely describe Fig. 4, in fact any one among the configurations of Fig. 5 fits sentence (23) since nothing is said concerning the relation between A^2 and A^3 .

To obtain an additional PO relation between A^2 and A^3 , we rewrite $r_{\{1\},2}$ in the following way:

⁴ We use a sans-serif font (as in NTPP) for logical predicates, and a roman font (as in NTPP) for r-items in the SR grammar.

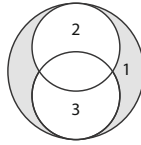


Fig. 4 First example of visual sentence

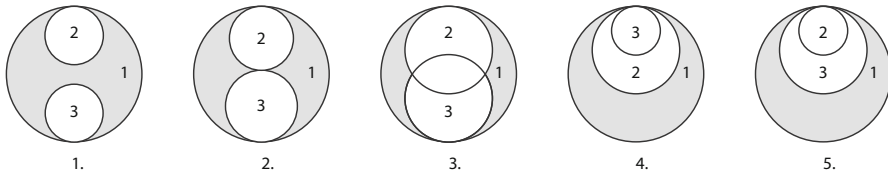


Fig. 5 Figures partly described by visual sentence (23), representing the five possible relations between A^2 and A^3 : $DC(A^2, A^3)$, $EC(A^2, A^3)$, $PO(A^2, A^3)$, $TPP(A^2, A^3)$, $TPP(A^3, A^2)$

$$r_{\{1,3\}}: @(*, \bigcirc) \rightarrow \{ @(*, \square), PO(*, \triangle) \}.$$

Now, derivation (22) becomes:

$$(21) \xrightarrow{r_{\{1,3\}}(TPP(A^2, A^1))} \langle \{A^3, A^4, A^2\}, \{TPP(A^2, A^3), TPP(A^4, A^3), PO(A^2, A^4)\} \rangle \tag{24}$$

and by renumbering superscripts, we get the configuration of Fig. 4.

Finally, we use standard s-productions and r-productions to rewrite nonterminals by terminals:

$$\begin{aligned} s_2: \quad \bigcirc &\rightarrow \langle \{a'\}, \emptyset \rangle \\ r_{\{1\},1}: \quad @(\bigcirc, *) &\rightarrow \{ @\{a', *\} \} \\ r_{\{1\},2}: \quad @(*, \bigcirc) &\rightarrow \{ @\{*, a'\} \}, \end{aligned}$$

where @ can be any relation, * any terminal or nonterminal, and a' denotes duly renumbered terminals.

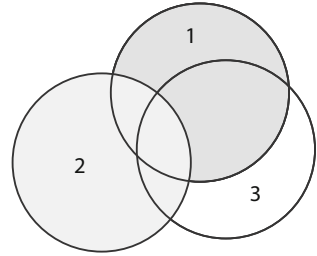
Let us now illustrate SR-RCC8 by another example: three regions having nonempty pairwise intersections but also a nonempty three-fold intersection (cf. Fig. 6).

We introduce s-productions and r-productions as in the first example:

$$\begin{aligned} s_2: \quad S &\rightarrow \langle \{\square, \triangle\}, \{PO(\square, \triangle)\} \rangle \\ s_3: \quad \bigcirc &\rightarrow \langle \{\square, \triangle\}, \{PO(\square, \triangle)\} \rangle \\ r_{\{3\},1}: \quad @(\bigcirc, *) &\rightarrow \{ @\{\square, *\} \} \\ r_{\{3\},2}: \quad @(*, \bigcirc) &\rightarrow \{ @\{*, \square\} \}, \end{aligned}$$

where * is any symbol.

Fig. 6 Second example of visual sentence



Let us start deriving:

$$S \xrightarrow{s_2(S)} \langle \{A^1, A^2\}, \{PO(A^1, A^2)\} \rangle \tag{25}$$

$$\xrightarrow{s_3(A^1)} \langle \{A^3, A^4, A^2\}, \{PO(A^1, A^2), PO(A^3, A^4)\} \rangle \tag{26}$$

$$\xrightarrow{r_{\{3\},1}(PO(A^1, A^2))} \langle \{A^3, A^4, A^2\}, \{PO(A^3, A^2), PO(A^3, A^4)\} \rangle . \tag{27}$$

As in the first example, we obtain a visual sentence that is incomplete. Indeed, it misses one of the three pairwise PO relations (and hence it also partly describes Fig. 7a, which is topologically distinct from Fig. 6). To obtain the third pairwise PO relation we introduce, as in the first example, a new r-production:

$$r_{\{3\},3}: @(\bigcirc, *) \rightarrow \{ @(\square, *), PO(\triangle, *) \} .$$

We now get the new derivation

$$(26) \xrightarrow{r_{\{3\},3}(PO(A^1, A^2))} \langle \{A^3, A^4, A^2\}, \{PO(A^3, A^2), PO(A^3, A^4), PO(A^4, A^2)\} \rangle . \tag{28}$$

But, contrarily to the first example, we are not done yet, because the visual sentence we obtain, namely (after renumbering)

$$\langle \{A^1, A^2, A^3\}, \{PO(A^1, A^2), PO(A^1, A^3), PO(A^2, A^3)\} \rangle , \tag{29}$$

does not guarantee a nonempty three-fold intersection—one may have pairwise nonempty intersections but not a three-fold one, as illustrated in Fig. 7b.

How can we guarantee the existence of a nonempty three-fold intersection? To attain that goal, we will introduce an additional constraint: the existence of a fourth (nonempty) region A^4 which is a nontangential proper part of all three A^1, A^2, A^3 . We rewrite rule s_3 as follows:

$$s_4: \bigcirc \rightarrow \langle \{ \square, \triangle, \nabla \}, \{ PO(\square, \triangle), NTPP(\nabla, \square), NTPP(\nabla, \triangle) \} \rangle$$

and add a new r-production

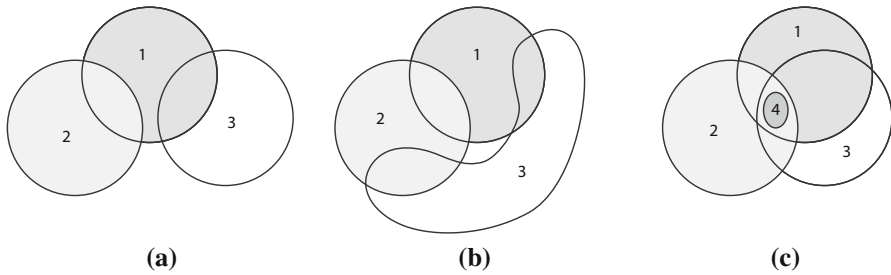


Fig. 7 **a** $\langle \{A^1, A^2, A^3\}, \{\text{PO}(A^1, A^2), \text{PO}(A^1, A^3), \text{DC}(A^2, A^3)\} \rangle$. **b** $\langle \{A^1, A^2, A^3\}, \{\text{PO}(A^1, A^2), \text{PO}(A^1, A^3), \text{PO}(A^2, A^3)\} \rangle$ but no nonempty three-fold intersection $A^1 \cap A^2 \cap A^3$. **c** $\langle \{A^1, A^2, A^3, A^4\}, \{\text{PO}(A^1, A^2), \text{PO}(A^1, A^3), \text{PO}(A^2, A^3), \text{NTPP}(A^4, A^1), \text{NTPP}(A^4, A^2), \text{NTPP}(A^4, A^3)\} \rangle$

$$r_{\{4\},1}: \text{PO}(\bigcirc, *) \rightarrow \{\text{PO}(\square, *), \text{PO}(\triangle, *), \text{NTPP}(\nabla, *)\}.$$

Let us derive a new:

$$S \xrightarrow{s_2(S)} \langle \{A^1, A^2\}, \{\text{PO}(A^1, A^2)\} \rangle \tag{30}$$

$$\xrightarrow{s_4(A^1)} \langle \{A^3, A^4, A^5, A^2\}, \{\text{PO}(A^1, A^2), \text{PO}(A^3, A^4), \text{NTPP}(A^5, A^3), \text{NTPP}(A^5, A^4)\} \rangle \tag{31}$$

$$\xrightarrow{r_{\{4\},1}(\text{PO}(A^1, A^2))} \langle \{A^3, A^4, A^5, A^2\}, \{\text{PO}(A^3, A^2), \text{PO}(A^4, A^2), \text{PO}(A^3, A^4), \text{NTPP}(A^5, A^3), \text{NTPP}(A^5, A^4), \text{NTPP}(A^5, A^2)\} \rangle. \tag{32}$$

After renumbering and replacing nonterminals by terminals, we get the desired visual sentence (cf. Fig. 7c):

$$\langle \{a^1, a^2, a^3, a^4\}, \{\text{PO}(a^1, a^2), \text{PO}(a^1, a^3), \text{PO}(a^2, a^3), \text{NTPP}(a^4, a^1), \text{NTPP}(a^4, a^2), \text{NTPP}(a^4, a^3)\} \rangle. \tag{33}$$

These examples show how SR-grammars introduce s-items and r-items either directly, by s-productions, or *contextually*, by r-productions. Indeed, even though SR grammars are context-free (in the sense that the left part of every rewrite rule consists of a single nonterminal), a theorem by Ferrucci et al. (1996, Theorem 3.1) states that *every contextual standard formal grammar can be written as a (context-free) SR-grammar*.

Let us see now how the semantics of a visual sentence are obtained by a bottom-up synthetic approach applied to semantic attributes. We will return to the first example: the syntax tree of the visual sentence corresponding to Fig. 4 can be seen in Fig. 8.

We attach semantic attributes (denoted by the index σ) to the nodes of this tree, starting by the leaves: the semantic attribute a^i_σ of a terminal s-item a^i will be a unary predicate $\text{Reg}(a^i)$ attesting that region a^i exists. The semantic attribute TPP_σ

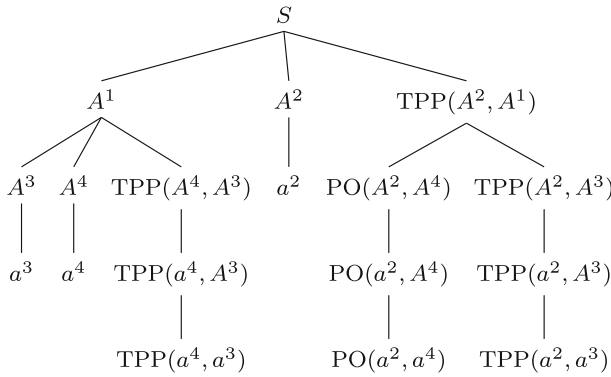


Fig. 8 Syntax tree of the visual sentence of Fig. 4

of TPP will be a binary predicate **TPP** attesting that its first argument is a tangential proper part of the second. Here are the semantic attribute synthesis rules that correspond to the s-productions and r-productions used for this example:

$$\begin{aligned}
 s_0: S_\sigma &\leftarrow \square_\sigma \wedge \triangle_\sigma \wedge \text{TPP}(\triangle, \square)_\sigma \\
 s_1: \bigcirc_\sigma &\leftarrow \square_\sigma \wedge \triangle_\sigma \wedge \text{TPP}(\triangle, \square)_\sigma \\
 r_{\{1\},1}: @(\bigcirc, *)_\sigma &\leftarrow @(\square, *)_\sigma \\
 r_{\{1\},2}: @(*, \bigcirc)_\sigma &\leftarrow @(*, \square)_\sigma \\
 r_{\{1\},3}: @(*, \bigcirc)_\sigma &\leftarrow @(*, \square)_\sigma \wedge \text{PO}(*, \triangle)_\sigma \\
 s_\lambda: \bigcirc_\sigma &\leftarrow a_\sigma \\
 r_{\{\lambda\},1}: @(\bigcirc, *)_\sigma &\leftarrow @(a, *)_\sigma \\
 r_{\{\lambda\},2}: @(*, \bigcirc)_\sigma &\leftarrow @(*, a)_\sigma,
 \end{aligned}$$

where @ is any relation and * any terminal or nonterminal. The reader can see in Fig. 9 the tree of semantic attributes of this example.

As expected, the semantic representation of Fig. 4 in first-order logic is (after renumbering):

$$\text{Reg}(a^1) \wedge \text{Reg}(a^2) \wedge \text{Reg}(a^3) \wedge \text{TPP}(a^2, a^1) \wedge \text{TPP}(a^3, a^1) \wedge \text{PO}(a^2, a^3). \quad (34)$$

The RCC8-based visual language we described in this section will be the basis of the visual part of INAUT, a hybrid language dedicated to maps and companion texts which we will describe in Sect. 5. But before that, let us see in the next sections how to combine visual and textual languages into hybrid languages (Sect. 3) and how to model maps and companion texts as hybrid language (Sect. 4).

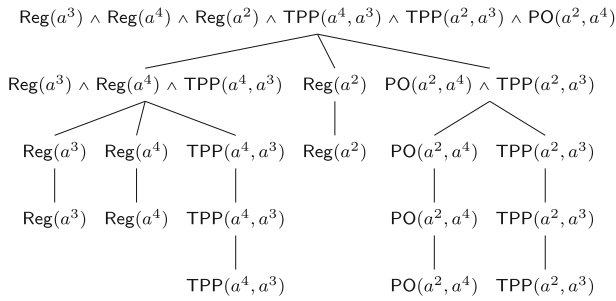



Fig. 9 Tree of semantic attributes of the visual sentence of Fig. 4


3 Controlled hybrid languages

3.1 An example

Let us start with a practical application example: consider the following map





extract:  , as a figure it can be modeled by a visual sentence in



the SR-RCC8 language described in the previous section. Now let us imagine that this map extract is accompanied by a sentence in a controlled natural language as follows:



[The wreck of Morania 130] lies at the bottom of [lake Erie].

(35)

The map extract consists of two visible objects:  and , as well as their legends. In the text we also have two objects (the brackets denote geolocated geographical entities): “[lake Erie]” and “[the wreck of Morania 130]”. Clearly,  and “[lake Erie]” are coreferential entities: their common referent is lake Erie (between Canada and the US)—similarly,  and “[the wreck of Morania 130]” corefer to the wreck of the Morania #130, a 120-feet freight barge that sank in 1951 and lies now at the bottom of lake Erie.



To describe the map extract in SR-RCC8 we will consider atomic figures  and  and their legends as individual terminal s-items. An r-item, which we will call λ (as in “lexical”), will make the link between visual entity and legend. The syntax of the visual sentence can be seen on the lower left part of Fig. 10. To this tree we can add also information that is contained in the electronic map, but is not necessarily visible, like the type of each object (here we have two types: *Wreck* and *Lake*), and its geographic coordinates. By extracting semantics from the syntax tree, we obtain a first-order logic formula part of the syntax tree of which can be seen in Fig. 10:


$$\text{Wreck}(\oplus) \wedge \text{Lake}(\text{Lake Erie}) \wedge \text{NTPP}(\oplus, \text{Lake Erie}) \wedge \lambda(\text{Lake Erie}, \text{“Lake Erie”}) \wedge \lambda(\oplus, \text{“Morania 130”}). \tag{36}$$

On the other hand, the textual sentence “[The wreck of Morania 130] lies at the bottom of [lake Erie]” gives rise to the syntax tree on the right lower part of Fig. 10. If we consider *M* and *E* the logical constants corresponding to the semantics of geolocated entities “[The wreck of Morania 130]” and “[lake Erie],” then the semantics we obtain from the textual sentence is

$$\text{lies}(M, \text{bottom}(E)) \tag{37}$$

where *lies* is a binary predicate and *bottom* a unary function.

After a coreference resolution phase, which will identify  and *E*, as well as  and *M* as being coreferent, we can merge logical formulas (36) and (37) into a first-order logical formula involving objects with attributes, the syntax tree of which is displayed on the top of Fig. 10.

As can be seen in this example, each modality carries specific information: the visual sentence provides a specific location of objects, their sizes, and shape in the case of the lake—the textual sentences provides the information that the wreck lies *at the bottom* of the lake, an information that the map contains only partly, by the choice of symbol , which, in fact, signifies “submerged wreck,” which is semantically slightly weaker than “lying at the bottom of”. On the other hand, when starting from the first-order formula at the top of Fig. 10, one can use several strategies to

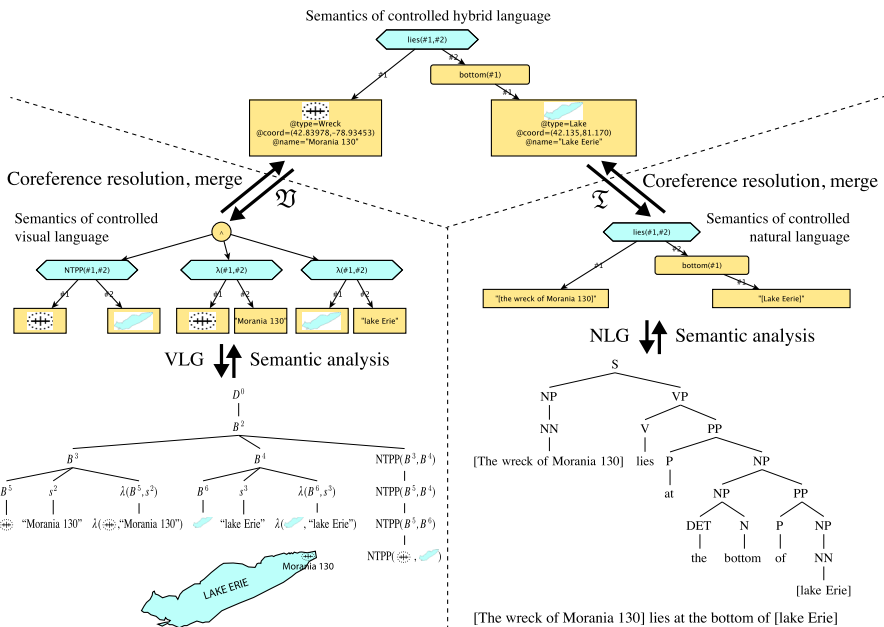


Fig. 10 Surface, syntax and semantics of the hybrid language sentence (35)

dispatch information into the two modalities, always keeping some coreferential redundancy to allow a combined reading of the two modalities by the human reader.

When generating hybrid language out of semantics represented in first-order logic formalism, besides a mechanism for dispatching information into the two modalities, one can also consider the possibility of adding exterior knowledge and perform inference. Let us first see how the logical formulas obtained from visual and textual sentences are merged in a KB graph.

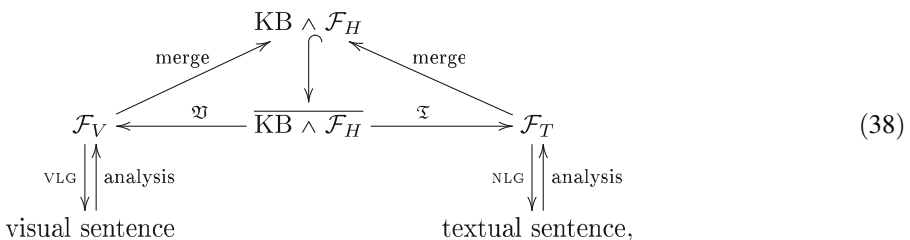
3.2 The knowledge-base graph

In the upper part of Fig. 10 the reader can see an example of graphical representation of first-order logical formula (where binary predicates relating an object with a property, such as type, coordinates and lexical reference, are represented as attributes). This representation is in fact a directed tree, the root of which is binary predicate *lies*, the outgoing edges of which correspond to its two arguments. The intermediate node *bottom* represents a function, and the two leaves are constants representing (separate) geolocated entities.

The syntax of a logical formula (or of a KB considered as the conjunction of several logical formulas) is canonically represented by a tree—the problem is that in this tree every occurrence of a constant representing the same geolocated entity is a *separate* individual node. We merge nodes representing the same geolocated entities. By doing this we lose acyclicity, but gain a more compact graph. After that, we remove high level conjunction operator nodes, considering that conjunction is the standard operation subsumed between connected components of the graph. In Fig. 11 the reader can see an example of the two-step process of converting a conjunction of logical formulas into a KB graph, as used in the INAUT system. An more extensive example is given in Fig. 14, on p. 20.

3.3 The principle of controlled hybrid language

The principle of *controlled hybrid language*⁵ can be illustrated by the following diagram:



⁵ Our notion of language hybridity differs from the one given in Costagliola et al. (1999), which deals with visual languages only. In Costagliola et al. (1999), a hybrid language is defined as a visual language using both logical (Plex syntactic model) and spatial relations (Box syntactic model).

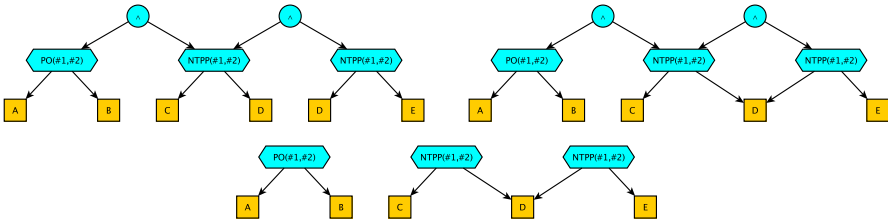


Fig. 11 From (the conjunction of) logical formulas $PO(A, B) \wedge NTPP(C, D) \wedge NTPP(D, E)$ to the KB graph, in two steps. *Upper left diagram* the original formula, upper right: merging leaves representing the same constants, *lower diagram* removing higher level conjunction operators

where KB is a knowledge base of “background” domain knowledge; \mathcal{F}_H is a first-order logic formula containing information and knowledge that covers both the visual modality and the textual modality; $\overline{KB \wedge \mathcal{F}_H}$ is the inferential closure of KB and \mathcal{F}_H ; \mathcal{F}_V is a first-order logic formula representing the semantics of the visual sentence; \mathcal{F}_T is a first-order logic formula representing the semantics of the textual sentence; \mathfrak{B} and \mathfrak{T} are filtering applications providing the contents of the visual sentence and of the textual sentence; “analysis” means obtaining semantics through the synthetic bottom-up approach of semantic attributes of the syntax tree; “NLG” means Natural Language Generation, and “VLG” means Visual Language Generation.

3.4 Related works

Because of the pyramidal structure of the controlled hybrid language architecture (38), one might be tempted to compare it with interlingua-based systems such as the Grammatical Framework (GF) (Ranta 2009; Angelov and Ranta 2009). Indeed, one can compare the formalism at the top of the controlled hybrid language diagram with GF’s abstract syntax, since it serves to produce two different “concrete syntaxes” (in the sense of GF). There are, nevertheless, two important differences between controlled hybrid languages and GF:

1. GF uses a single method for going from abstract to concrete syntax, namely linearization. By changing linearization rules, one obtains various different textual outputs, but the linearization mechanism is the same. This mechanism can be compared with our bottom-up attribute synthesis approach, but for standard grammars only. Controlled hybrid languages use different approaches for textual and for visual sentences: standard grammars in the former case, SR-grammars in the latter.
2. While GF will independently produce output in different languages, controlled hybrid languages produce bimodal output simultaneously and the issue of dispatching information between the two modalities is of uttermost importance. In other words, an hybrid language is necessarily a mixture of a textual and a visual language, while GF’s output in each language is intended to be self-contained.

Nevertheless it would be interesting to build a GF implementation with visual language linearization features and with an intelligent dispatching module between the two modalities.

4 Geographic maps with companion texts as sentences in controlled hybrid language

A geographic map is a representation of a fragment of the Earth's surface (whether land or sea, or both) using elements that sometimes mimic geographical entities (for example a sea coast line or the shape of a town) and sometimes have symbolic shapes with specific semantics (for example there are special symbols for light towers or for ship wrecks). These two functions can be combined; for example, the stroke representing a highway is mimicking the stroke of real-world highway and has shape (thickness, parallel lines, etc.) that carries specific semantics to distinguish it from other kinds of roads. To facilitate reading of the map, map designers include a box (called a *legend*) presenting representative symbols and their semantics.

A *companion text* of a map is any text complementing the map—and hence using the same terms as the map legend and the same toponyms as the map. It is usually produced by the same entity that produced the map and one can assume that the information it contains is semantically consistent with the information contained in the map.

When representing geographic maps with companion texts as controlled hybrid language, types and objects are shared by the map and the companion text. Relations between objects can be those of topological logic described in Sect. 3 but also relations of other nature (for example, the entrance and the exit of a channel are related by the fact that they belong to the same channel).

Attributes can be of various kinds, but two of them are omnipresent: `coord`, containing geographic coordinates of the referent and `name`, containing the standard name of the referent in a given language and script combination.

S-items will be visual elements of various kinds. According to Schlichtmann (1984), we have a taxonomy of visual symbols given in Fig. 12: locational information is about a specific location (a point in space) while substantive information has shape carrying semantics; this shape can be a scaled-down reproduction of a real-world element (plan information) or can be symbolic (plan-free information). And in the case of symbolic elements, one can still use several parameters carrying meaning: shape, size, color value, texture, hue, orientation (Bertin 1999).

R-items allow to link map objects between them to encode essential properties of map objects and to avoid inconsistencies such as displaying only the entry of a channel and not its exit. As usual, a visual sentence will be a set of s-items and a set of r-items using the s-items. Note that in the case of electronic maps, since the geographic coordinates of every instance that is visible on the map are included in the KB (as values of the `coord` attribute) the precise position of the representation of each instance is obtained by converting its geographic coordinates into the coordinate system of the map. Therefore one has both quantitative and qualitative


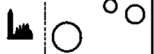

Contents (Information)	Expressions	Spatial Characteristics	Non-spatial Characteristics
	Locational information		+ +
	Plan information		✕
Substantive information	Plan-free information		

Fig. 12 The taxonomy of visual elements encountered in maps as given by Schlichtmann (1984, p. 24) cited by Grant Head (1991, p. 247)

spatial information: the former is made of coordinates of shapes represented by closed polylines, and the latter is based on RCC8. In many cases the latter is obtainable from the former by spatial inference—but whether it is automatically obtainable or not, it is very useful both on the textual side (for text generation, content determination and sentence ordering) and on the visual side (for interaction with the pointing device).

After having discussed geographic maps with companion texts in general as controlled hybrid languages, let us turn to the description of the specific project: a controlled hybrid language for the French nautical charts and the companion *Instructions nautiques*.

5 A controlled hybrid language for the French nautical charts and their companion texts

During the last centuries, several French institutions have been in charge of creating and maintaining “official” maritime charts, starting with the *Dépôt des cartes et plans de la Marine* founded in 1720 by Louis XV. Since 1971, the French Naval Hydrographic and Oceanographic Service (SHOM) is publishing nautical charts (both on paper and as ENCs) and companion books (nautical instructions, tidal almanacks, signal books, etc.). These are optional for pleasure sailing but mandatory for commercial and French Navy ships.

Instructions nautiques is the name of a nautical book series⁶ published by the SHOM. They are the French counterpart of the *United States Coast Pilot*⁷, published by the United States National Oceanic and Atmospheric Administration’s Office of Coast Survey, and of the *British Admiralty Sailing Directions*⁸ published by the United Kingdom Hydrographic Office.

⁶ <http://www.shom.fr/les-produits/produits-nautiques/ouvrages-nautiques/instructions-nautiques/>.

⁷ <http://www.nauticalcharts.noaa.gov/nsd/cpdownload.htm>.

⁸ <https://www.ukho.gov.uk/PRODUCTSANDSERVICES/PAPERPUBLICATIONS/Pages/NauticalPubs.aspx>.

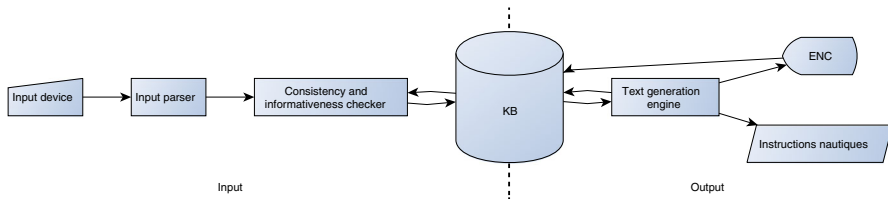


Fig. 13 Architecture of the SHOM KB system

Information for the *Instructions nautiques* is provided by survey vessels, port officers, maritime officers and mariners in general. In some cases, this information may require immediate update, for example to notify a shipwreck or some important change of navigation conditions—in other cases it may remain unchanged for decades. Also *Instructions nautiques* can be used both as separate documents and as information providers for interactive chart display devices, to supply on-the-fly information that is specific to an area or object selected by the user, and evolving according to the current navigation context.

To alleviate these needs, SHOM is building a knowledge base that will cover both ENCs and their companion texts. This knowledge base will be used for generating the *Instructions nautiques* and also to communicate with ENCs and navigation equipment for on-the-fly text generation. But how can mariners, not necessarily proficient in knowledge base management and ontologies, supply additions and corrections to this knowledge base?

This was our motivation for developing INAUT: a controlled hybrid language, the textual part of which is based on French natural language and the visual part of which is based on ENC objects and relations. INAUT is designed for human-to-machine communication: mariners will enter text in INAUT on a specific GUI and submit it to the KB. When going in the opposite direction (machine-to-human), the system will generate either *Instructions nautiques* fragments of small text excerpts to be displayed on ENCs. The reader can see the SHOM KB system's architecture in Fig. 13.

To our knowledge, INAUT is the first maritime controlled language.⁹

5.1 A sample paragraph in INAUT language

In Fig. 14, the reader can see a fragment of a (slightly simplified¹⁰) KB graph based on Vol. D2.1 § 2.2.4 of the *Instructions nautiques* and on nautical chart FR57003C of the SHOM. To submit the data to the KB, one would need to enter the following INAUT text, while selecting in the corresponding ENC fragment (shown in Fig. 15) the regions that are marked by brackets in the text:

⁹ With the exception of Seaspeak (Stevens and Johnson 1983), a controlled natural language defined in 1985 by the International Maritime Lecturers Association. In 2001 it evolved into SMCP (Standard Marine Communication Phrases IMO 2005) which is still used today. These controlled natural languages, dedicated to oral communication between ships, are “human-oriented” (of type c in Kuhn 2014, p. 125).

¹⁰ We have omitted `coord` attributes and used `name` attribute values as node identifiers.

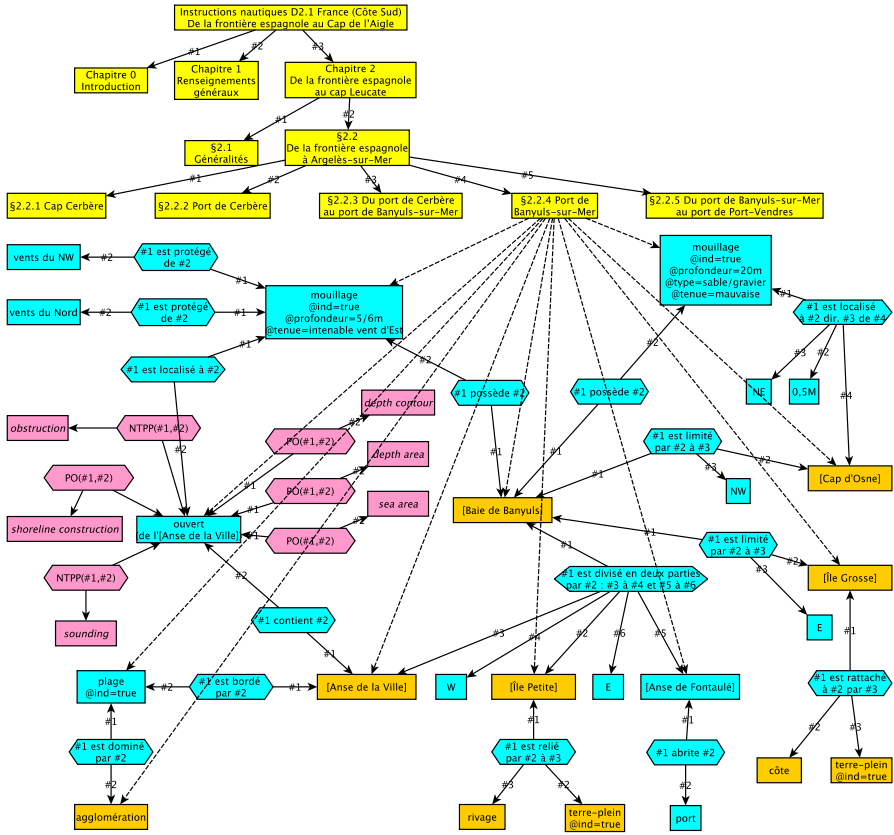


Fig. 14 Sample KB fragment. Nodes in *blue* appear only in the text, in *pink* only on the map, in *orange* both on the map and in the text. Nodes in *yellow* represent the hierarchical structure of the document. They are connected by *dotted arrows* to predicates belonging to the same section. Attributes are displayed inside object nodes. (Color figure online)

La [baie de Banyuls] est limitée au NW par le [cap d’Osne]. La [baie de Banyuls] est limitée à l’Est par l’[île Grosse]. L’[île Grosse] est rattachée à la côte par un terre-plein. La [baie de Banyuls] est divisée en deux parties par l’[île Petite]: à l’Ouest l’[anse de la Ville] et à l’Est l’[anse de Fontaulé]. L’[île Petite] est reliée au rivage par un terre-plein. L’[anse de la Ville] est bordée par une plage. La plage est dominée par l’agglomération. L’[anse de Fontaulé] abrite le port. La [baie de Banyuls] possède deux mouillages. Le premier mouillage est localisé au NE du [Cap d’Osne]. Le premier mouillage a une profondeur de 20 mètres. Le premier mouillage est de type sable et gravier. Le premier mouillage a une mauvaise tenue. Le deuxième mouillage est localisé à l’ouvert de l’[Anse de la Ville]. Le deuxième mouillage a une profondeur de 5 à 6 mètres. Le deuxième mouillage est protégé des vents de Nord. Le deuxième mouillage est protégé des vents de Nord-Ouest. Le deuxième mouillage est intenable par vents d’Est.

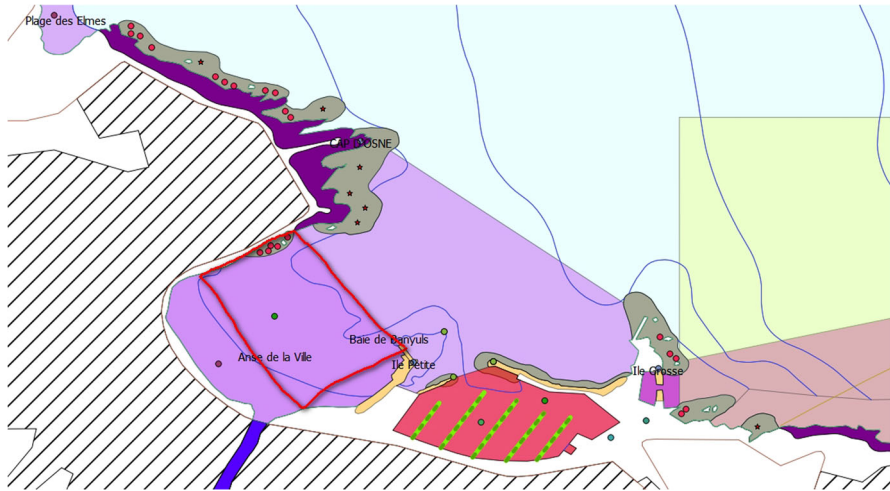


Fig. 15 The visual sentence (map) corresponding to the KB diagram of Fig. 14

This text contains many repetitions of the same subject (“le premier mouillage,” “le deuxième mouillage,” etc.) as well as cases where the direct object of one sentence is the subject of the next one. When generated from the KB, INAUT output based on the same KB subgraph takes the following more readable form:

La [baie de Banyuls] est limitée au NW par le [cap d’Osne] et à l’Est par l’[île Grosse] rattachée à la côte par un terre-plein. Elle est divisée en deux parties par l’[île Petite] reliée au rivage par un terre-plein : à l’Ouest l’[anse de la Ville] bordée par une plage dominée par l’agglomération et à l’Est l’[anse de Fontaulé] qui abrite le port. La [baie de Banyuls] possède deux mouillages : le premier a une profondeur de 20 mètres, il est de type sable et gravier, et a une mauvaise tenue ; le deuxième est localisé à l’ouvert de l’[Anse de la Ville], il a une profondeur de 5 à 6 mètres, il est protégé de vents de Nord et de Nord-Ouest, mais est intenable par vents d’Est.

5.2 The structure of INAUT



To describe INAUT as controlled hybrid language, we need to provide its types, objects, attribute keys and values, relations, lexical and visual references, as well as the grammars of the textual language and of the visual language. Let us start with the types.

5.2.1 INAUT types

There are two kinds of types used in INAUT: those of the International Hydrographic Organization (IHO) Special Publication S-57, and those reflecting the hierarchical structure of *Instructions nautiques*.

IHO S-57. The 1992 standard IHO S-57¹¹, entitled *IHO Transfer Standard for Digital Hydrographic Data* is used primarily for electronic navigational charts. It includes an abstract data model based on objects and relations. Objects can have attributes, and can be geolocated using three kind of geometric primitives: points, lines and areas.

For example, the object *Wreck* (acronym *WRECKS*) is defined as “the ruined remains of a stranded or sunken vessel which has been rendered useless,” and has 19 attributes, like *CATWRK* (category of wreck: it can take values 1 = “non-dangerous wreck,” 2 = “dangerous wreck,” 3 = “distributed remains of wreck,” 4 = “wreck showing mast(s),” and 5 = “wreck showing any portion of hull or superstructure,”) and *SCAMIN* (the minimum scale at which the object may be used for Electronic Chart Display and Information System presentation, an integer ≥ 1).



S-57 objects are included into INAUT’s type set. Their lexical references are standard translations of their official English names into French language (in the case of the *Wreck* concept it will be “épave”). The visual reference of *Wreck* concept instances is symbol  or symbol , depending on the value of attribute *CATWRK* (SHOM 2012, K-28, p. 39).

Hierarchical structure of Instructions nautiques. Every volume of the *Instructions nautiques* series has a standard hierarchical structure (Menanteau 2011, 2013). For example, volume D2.1 deals with Southern French coasts and is entitled “France (Côte Sud). De la frontière espagnole au Cap de l’Aigle”. All volumes start with a Chapter 0 entitled “Introduction,” and a Chapter 1 entitled “Renseignements généraux,” followed by chapters corresponding to large subdivisions of the coast covered by the volume. Chapters are subdivided in sections and sections in subsections, covering nested parts of the coast. Every subsection is subdivided in parts called “Généralités,” “Atterissage,” “Mouillages,” etc. depending on the specific case of coast fragment.

When generating a volume of the series, it is essential to know to which subsection part belongs each textual instance. For this reason we introduce types *Volume*, *Chapter*, *Section*, *Subsection*, the objects of which are hierarchical subdivisions of a given volume. For similar reasons we also introduce the type *Map*, the objects of which are used to specify the SHOM chart to which belongs each visual instance.

5.2.2 INAUT objects, attributes and relations

Most objects of S-57 types refer to real-world objects. From their name attributes, we obtain lexical references as used in text or in maps.

When generating text, attributes become adjectives (“un amer rouge”) or subordinate clauses (“un amer de couleur rouge”). In the case of image generation, symbols can be modified depending on attribute value (for example, a submerged wreck will be represented by symbol  while a partially visible one by symbol , the difference between the two being stored as value of the attribute *CATWRK*).

¹¹ http://www.iho.int/iho_pubs/IHO_Download.htm.

Lexical references of INAUT relations will usually be verbs (sometimes surrounded by grammatical words like prepositions). By inverting the order of edges between relations and objects we can switch from active to passive voice.

5.2.3 INAUT's textual grammar

INAUT is a controlled language with a large vocabulary (largely based on the existing *Instructions nautiques* corpus) and a rather simple syntax. Here is a (simplified) version of part of its grammar:

S → NP VP
 NP → DET NN | NN
 NN → ADJ NN | NN ADJ NOUN
 VP → VERB NP | VERB NP PP
 PP → PREP DET NN | PREP NN

where symbols in small caps are terminals.

The VERB, always in 3rd person or in the infinitive, can be active or passive. In most cases it is possible to change the voice of the verb, which implies a permutation of the NPs in SUBJECT and OBJECT position, leaving the PPs intact:

La [baie de Banyuls] est limitée par le [cap d'Osne] au NW.

Le [cap d'Osne] limite la [baie de Banyuls] au NW.

Definite articles are used for all objects the names of which start with the name of the type to which the object belongs: for example, the name “baie de Banyuls” starts with “baie” (=bay) which is the lexical reference of an INAUT type, hence in INAUT the definite article is used: “la [baie de Banyuls]”.

Otherwise, no article is used:

[Notre-Dame de la Salette] est un amer remarquable à l'WSW du port.

Non geolocated instances are, by default, used with definite articles. When an indefinite article is required, the information is stored in a dedicated attribute *ind*. Indefinite articles are used in OBJECT position only:

L'[anse de la Ville] est bordée par *une* plage. *La* plage est dominée par l'agglomération.

5.2.4 INAUT's visual grammar

INAUT's visual grammar is an SR-grammar as described in Sect. 2.1. S-items are ENC objects and hence are taken from S-57 as described above (Sect. 5.2.1). R-items follow RCC8 (cf. Sect. 2.2) but also include lexical references (as in Sect. 3.1), coordinates and other relations based on the (relatively sparse) set of relations provided by S-57. Figure 10 shows an example of a (simplistic) INAUT visual sentence, its syntax tree and semantics.

5.3 Complementarity of visual and textual representations in INAUT

Besides the properties of intermodal interaction and sharing information between modalities that are common to all hybrid languages, the specific case of INAUT has an additional very interesting property: there is a complementariness between textual and visual information due to the fact that the agent's point of view and intention are different. Take for example the sentence

La plage est dominée par l'agglomération.

The verb “dominer” should not be understood as a graphical relation between objects “agglomération” and “plage” in the map, but as a visual relation of the two real-world referents in the landscape as viewed by a ship entering the bay of Banyuls. Indeed, a large part of landscape descriptions in *Instructions nautiques* is intended as an *aid for navigators to visually locate their position* (since GPS geolocation often proves insufficient). In the case of the example sentence, the reader can see in Fig. 16 how the information provided by the sentence reflects reality.

Following these considerations one could envision a 3D extension of the visual part of the INAUT language where 2D information is generated from the map objects, while information on 3D aspects of objects is obtained from the text.

5.4 INAUT textual language generation

As mentioned already, INAUT will be used both for generating entire *Instructions nautiques* volumes as well as small paragraphs of text to be displayed in ENC visualization and navigation devices. This is typically a Natural Language Generation problem.

Reiter and Dale (2000, § 3.3) divide the language generation task into seven subtasks: content determination, document structuring, lexicalization, aggregation, referring expression generation, linguistic realization and structure realization.

5.4.1 Content determination

Considering the KB as a graph (cf. Sect. 3.2), the problem of content determination becomes a KB subgraph selection problem, according to three criteria: geolocation, hierarchical structure, and predicate-based closure.

Indeed, there are several ways of initiating a text generation process in INAUT: by selecting a hierarchical structure node in the knowledge base (in case one wants to produce a fragment of *Instructions nautiques* volume), by selecting one or more map objects by drawing a zone with the pointing device on the level of the GUI of the ENC, etc. In all cases, whether the selection is based on hierarchical structure or is done manually at the level of the GUI of the ENC, we obtain a first set of selected geolocated nodes. Algorithm 1 (cf. p. 1) calculates, out of this set, the *predicate-based closure of the subgraph*, and returns the subgraph which will serve as content for the natural language generation process. In the algorithm we use the following

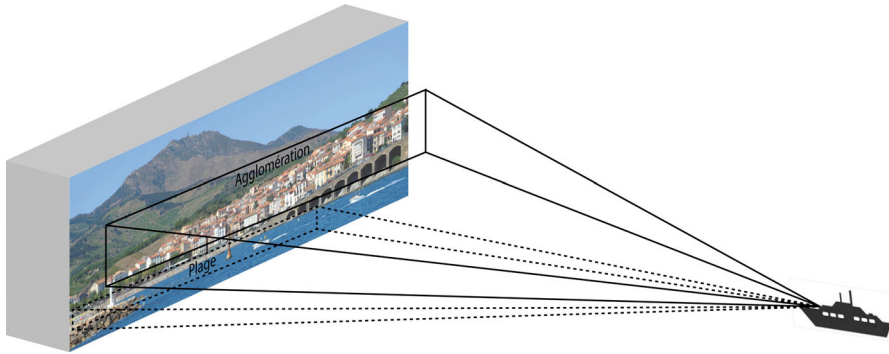


Fig. 16 Intention of the sentence “La plage est dominée par l’agglomération” (=the beach is dominated by the town) as description of the landscape from the point of view of a vessel approaching baie de Banyuls

notations: \mathcal{N} is the set of nodes corresponding to logical constants, \mathcal{P} the set of predicate nodes, and if $P(X, Y)$ is a binary predicate we denote by $\text{Args}(P)$ the set of its arguments $\{X, Y\}$ and for each $A \in \text{Args}(P)$ let $\text{edge}_P(A)$ be the edge connecting P to A in the graph. We also denote by $\text{Branch}(v)$ (where v is a vertex) the induced subgraph of all (directed) descendants of v (including itself) and edges connecting them (this will be useful for logical functions).

The algorithm can be described as follows: (1) from the set \mathbf{X} of initially selected nodes we obtain the set of predicates \mathbf{P} involving some initial node, either directly or inside a term; (2) we connect the elements of \mathbf{P} with their arguments (if an argument is a function then we retrieve the whole directed branch under it). This means that we get all predicates connected to the initial nodes and then all nodes connected to the predicates, we call it, a “predicate-based closure” of the subgraph.

```

Input:  $\mathbf{X} \subset \mathcal{N}$  the set of initially selected nodes
Result: The subgraph  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$  which will be used for text generation
 $\mathbf{P} \leftarrow \emptyset, \mathbf{V} \leftarrow \emptyset, \mathbf{E} \leftarrow \emptyset;$ 
for  $X \in \mathbf{X}$  do
  for  $P \in \{P \in \mathcal{P} \mid (X \in \text{Args}(P)) \vee (f_1 \circ \dots \circ f_i(\dots, X, \dots) \in \text{Args}(P))\}$  do
     $\mathbf{P} \leftarrow \mathbf{P} \cup \{P\};$ 
  end
end
 $\mathbf{V} \leftarrow \mathbf{V} \cup \mathbf{P};$ 
for  $P \in \mathbf{P}$  do
  for  $A \in \text{Args}(P)$  do
    if  $A$  is a logical function then
       $\mathbf{G} \leftarrow \mathbf{G} \cup \{\text{Branch}(A)\}$ 
    else
       $\mathbf{V} \leftarrow A;$ 
    end
     $\mathbf{E} \leftarrow \mathbf{E} \cup \{\text{edge}_P(A)\};$ 
  end
end

```

Algorithm 1: Content determination algorithm.

5.4.2 Document structuring, aggregation and linguistic realization

This is a difficult phase since it deals with linearizing a fragment of the KB graph by providing the order in which sentences are written. Legacy NLG systems order sentences by considering the various types of discourse relations between them (Reiter and Dale 2000, § 4.4.1), but this method cannot be applied in our case because the discourse relations between INAUT sentences are of a single type, namely elaborations (Mann and Thompson 1988, p. 273). Therefore we derive the order of sentences rather from their context, by using machine learning methods. The approach proposed by Cohen et al. (1999) suits our needs¹² as it takes the context into account by using a *preference function*. A set of experts is asked to order elements of a set which allow us to learn the preference function. This binary function can be noted as $P(x, y)$ where the output is a measure of confidence displaying the preference of x over y . This preference is learned from experts first ordering and users feedback on the order of sentences.

For future work, we can cite Roth and Frank (2010) that uses an Expectation–Maximization based algorithm in order to align geographical route representations with natural language directions. While it doesn't address the issue of planning, the machine learning technique presented could be used to compare alignments in our system between the geographical path in the document (cf. infra) with the generated text to evaluate the quality of the generation.

Before continuing, let us note the existence of a geographical path which is implicit to the hierarchical structure of the document: indeed, in every volume of the *Instructions nautiques*, sections describe fragments of coast line ordered in a given direction, so that object centers are roughly located on a path. We call this, the *guiding path* of the volume (the reader can see an example in Fig. 18).

Let \mathbf{G} be the subgraph of KB obtained as output of Algorithm 1. We subdivide the document structuring task into four subtasks:

1. sort connected components \mathbf{G}_i of \mathbf{G} ;
2. find a starting node S for each component;
3. establish a split and an order of sentences;
4. convert relations into INAUT.

Subtask 1: Sort connected components of \mathbf{G} . The criteria for sorting components are: (a) if there is a significant difference in size between the cumulative geographic areas of two components, the larger one will precede the smaller one, (b) otherwise, calculate the barycenters of cumulative geographic areas of components; the path defined by their barycenters should be roughly parallel to the guiding path of the volume.

Subtask 2: Find a starting node S for each component. In the case where text generation starts with the selection of a map object by the user, this step is trivial: the starting node is the one obtained by the selected map object.

¹² Although, it is noteworthy to mention the approach of Freund et al. (1998) that uses a boosting algorithm, *RankBoost*, to learn to order a set of elements encoded in terms of a set of features depending on the learning corpus and which is used, inter alia, by Walker et al. (2001) to train a sentence planner.

In the case of *Instructions nautiques* text generation, we define a ranking of **G** nodes in order to determine *S* as the highest ranked node according to the following criteria:

1. The first criterion for choosing *S* is the similarity of lexical representation of nodes $X \in \mathbf{G}$ with those of the hierarchical structure nodes. For example, in Fig. 14, if the lexical reference of section “Sect. 2.2.4,” contains the string “Port de Banyuls-sur-Mer” and the node *X* with lexical reference “[Baie de Banyuls]” is lexically closest to it.

2. A second criterion can be node degree (by construction, geographic object nodes are connected with predicates (sometimes via functions), and chances are the node with the most relations is the most important of a given area).

3. Nodes that appear both on the map and in the *Instructions nautiques* are considered more important than those appearing only in one modality.

4. Finally, another criterion is of semantic nature, the one of “interest” for the navigator: a ranking is established between concepts to which instances of **G** belong, for example a **PORT** instance will be more interesting than a **BEACH** instance. For some relations this weight is inherited by neighboring nodes: for instance, a bay containing a port is more interesting than a bay containing only a beach, etc. This allows the use of PageRank-like algorithms for finding the most important node in the subgraph.

These criteria are used as features for a machine learning process based on the existing *Instructions nautiques* corpus, to build a node ranker.

Subtask 3: Establish a split and an order of sentences. If we consider **G** to provide a paragraph by natural language generation, we have to split it into an ordered set of sentences (see also Sauvage-Vincent et al. 2015). There are two main issues involved: (a) in which order to produce sentences, and (b) how to delimit sentences. Indeed, concerning (b), there is a trade-off between building long complex sentences with many subordinate clauses (corresponding to relations at distance 3 and nodes at distance 4 from the starting node of the sentence) and splitting the paragraph into many short sentences with coreferent nouns far distanced.

To solve (a) we have established a ranking of relations for a given type of instance (for example, when applied to an area, “est limité” > “est divisé” > “possède,” the rationale being that one will first delimit an object, then describe its structure and then list its sub-objects). This ranking depends on the nature of relation but also on the type of arguments of the relation (“possède” to “mouillage” > “possède” applied to “accès”, Menanteau 2013, § 7.2).

Concerning (b) one has to apply a strategy whenever an object at distance 2 from *S* belongs to relations at distance 3 and beyond. Between the extreme strategy of including all pair distance objects in a single sentence using subordinate clauses:

La [Baie de Banyuls]₀ est divisée₁ en deux parties par l'Île Petite₂ qui est reliée₃ au rivage₄ par un terre-plein₄ : l'Anse de la Ville₂ qui est bordée₃ par une plage₄ dominée₅ par l'agglomération₆ et contient₃ l'ouvert₄ de l'Anse de la Ville où se trouve₅ un mouillage₆, à W₂, et l'Anse de Fontaulé₂ qui abrite₃ le port₄, à E₂,

(in this example, index number denotes the distance from *S*, italics denote subordinate clauses) and the equally extreme strategy of stopping at distance 2, and

placing all distance 2 objects with further descendants on a queue and then producing additional sentences by dequeuing them (which roughly corresponds to a Depth-First Search traversal):

La [Baie de Banyuls]₀ est divisée₁ en deux parties par l' [Île Petite]₂ : l' [Anse de la Ville]₂ à W₂ et l' [Anse de Fontaulé]₂ à E₂. L' [Île Petite]₂ est reliée₃ au rivage₄ par un terre-plein₄. L' [Anse de la Ville]₂ est bordée₃ par une plage₄. La plage₄ est dominée₅ par l' agglomération₆. L' [Anse de la Ville]₂ contient₃ l' ouvert₄ de l' [Anse de la Ville]. L' ouvert₄ de l' [Anse de la Ville] contient₅ un mouillage₆,

one can adopt intermediate strategies, like for example building subordinate clauses only when the distance 4 object does not participate in further relations. In our case this would give:

La [Baie de Banyuls]₀ est divisée₁ en deux parties par l' [Île Petite]₂ *qui est reliée₃ au rivage₄ par un terre-plein₄* : l' [Anse de la Ville]₂ à W₂ et l' [Anse de Fontaulé]₂ *qui abrite₃ le port₄*, à E₂. L' [Anse de la ville]₂ est bordée₃ par une plage₄ *dominée₅ par l' agglomération₆*. L' [Anse de la ville]₂ contient₃ l' ouvert₄ de l' [Anse de la Ville] *où se trouve₅ un mouillage₆*.

Nevertheless even this strategy is suboptimal since the relation “est divisée en deux parties par” implies a symmetry between the two parts, and this (geographical) symmetry is (syntactically) broken when we use subordinate clauses for one of the two parts and not for the other.

In fact, finding the right paragraph split is a complex problem depending on syntax, semantics, style and legacy conventions. We apply a hybrid approach aggregating a rule-based classifier (based on rules similar with the ones given above) and a machine-learning one (based on the existing *Instructions nautiques* corpus).

Note that the direction of paths connecting *S* to other nodes determines, in most cases, the voice of verbs used in the textual representation: indeed, the system considers that voice change does not alter semantics and hence, for example, no distinction is generally made between the following two sentences:

L' [Anse de la Ville] est bordée par une plage, dominée par l' agglomération.

L' agglomération domine une plage, qui borde l' [Anse de la Ville].

However, even though they are semantically equivalent, they are stylistically different, and the paragraph splitting strategy can potentially take verb voice as a feature when the classifier is trained from *Instructions nautiques* data.

Subtask 4: Finalize sentences. This subtask, executed at the same time as subtask 3, and called “microplanning” by Reiter and Dale (2000), mainly concerns referring expression generation. For example,

L' [Anse de la ville]₂ est bordée₃ par une plage₄ dominée₅ par l' agglomération₆. L' [Anse de la ville]₂ contient₃ l' ouvert₄ de l' [Anse de la Ville] où se trouve₅ un mouillage₆.

will become

L'[Anse de la ville]₂ est bordée₃ par une plage₄ dominée₅ par l'agglomération₆, et contient₃ l'ouvert₄ de l'[Anse de la Ville] où se trouve₅ un mouillage₆.

In some cases a list structure is used to represent sentences describing similar objects related to the same parent with similar relations. For example, the following (already aggregated) paragraph:

La [baie de Banyuls] possède deux mouillages. Le premier mouillage est localisé au NE du [Cap d'Osne], a une profondeur de 20 mètres, est de type sable et gravier et a une mauvaise tenue. Le deuxième mouillage est localisé à l'ouvert de l'[Anse de la Ville], a une profondeur de 5 à 6 mètres, est protégé des vents de Nord-Ouest et est intenable par vents d'Est,

will become

La [baie de Banyuls] possède deux mouillages:

1. localisé au NE du [Cap d'Osne], il a une profondeur de 20 mètres, est de type sable et gravier et a une mauvaise tenue;
2. localisé à l'ouvert de l'[Anse de la Ville], il a une profondeur de 5 à 6 mètres, est protégé des vents de Nord-Ouest et est intenable par vents d'Est.

In the case where (three or more) similar objects having exactly the same attributes are listed, one can also use tables, especially when attribute values are simple (numeric or short strings). Nevertheless, to comply with the style of legacy *Instructions nautiques*, tables are always based on pre-existing templates.

5.5 Classification

In the PENS classification scheme (modulo the fact that INAUT is French-based while Kuhn 2014 considers only English-based languages), the textual part of INAUT language can be classified as being $P^5E^3N^4S^3$ FWAG: P^5 because it is fully formal and specified on both syntactic and semantic levels; E^3 because it has relations of arity greater than 1, general rule structures and negation (its semantics are represented in first-order logic); N^4 since although complete documents are written in INAUT (namely the *Instructions nautiques*), it can only be used for describing landscapes and navigation conditions, and many basic features of natural language, such as questions or use of the first person, are unavailable; S^3 because of the large vocabulary of INAUT, the fact that concepts include the complete set of S-57 objects, etc.—in other words, it is a simple language but still quite lengthy to describe because of the many nouns and verbs it uses; F because it is used to “provide a natural and intuitive representation for formal notations”; w because it is intended to be written; and AG because it originates from an academic research project financed by a government agency (the SHOM).

According to Kuhn (2014) there are five other languages in the same $P^5E^3N^4S^3$ class:

1. *First Order English* (no more information available) (Pool 2006);
2. *Gherkin*, a language for writing executable scenarios for software specifications (example: “Given I am a student. And a lecture “PA042” with limited capacity of 20 students. But the capacity of this course is full.”) (Nečas 2011);
3. *iLastic Controlled English*, a language for writing intuitive and natural scripts (example: “delete all files under the tmp folder if the space of the disk is lower than 1024”)¹³;
4. *PENG*, a rich but unambiguous language that can be automatically translated via discourse representation structures into FOL with equality (example: “while the fox sleeps, the cat chases a bird”) (Schwitter 2002);
5. *RECON*, a language to represent fact and rules in an industrial environment with a deterministic mapping to FOL (example: “if any container contains part of a shipment, it contains no other shipment”) (Barkmeyer and Mattas 2012).

As for the visual part of INAUT, and although generally extending PENS to visual languages goes far beyond the scope of this paper, we can dare a tentative extension by simply leaving out the N part.¹⁴ Hence, we state that the visual part of INAUT can be classified as $P^5E^2S^3$: P^5 because the language is fully formal and fully specified; E^2 because we have relations of arity greater than 1 but no general rule structures and no logical operators; S^3 for the same reasons as for the textual part of INAUT as well as the fact that SR grammar descriptions are quite lengthy.

After extending PENS to visual languages, the next step will be to extend it to hybrid languages, and take into account the amounts of interaction and of complementariness between textual and visual languages.

5.6 Applications

5.6.1 Interaction with ENCs

As said in the introduction, *Instructions nautiques* are defined as companion texts to charts, and, in particular, to ENCs. Therefore it is important to define interactions between INAUT and ENCs. By choosing, for example, an object on an ENC the user may receive INAUT text in return. Generating this text automatically has the advantage of being (a) limited to the object given by the user; (b) adapted to local conditions, for example time of the day (some relations or attributes in KB may be time-dependent) or meteorological conditions, or parameters of the user’s vessel (size, tonnage, etc.); (c) up-to-date, since users may constantly provide new information.

Additional information can be added to the message sent to the KB by the ENC device, so that text can be filtered and only specific types of information displayed, as for example information on mooring, landing, etc.

¹³ <http://www.crunchbase.com/organization/ilastic>.

¹⁴ In PENS, the naturalness criterion refers to the relation of a controlled language to the natural language it is based upon. Even if we consider controlled visual languages as being based, in some sense, on more general “natural” visual languages (as discussed, for example, in Machin 2007), similarly to the way architectural (“formal”) blueprints are based on the architect’s primordial (“natural”) sketches, we consider that such a relation would be much more difficult and ambiguous to describe than in the case of textual languages, and hence is not necessarily appropriate for a general classification scheme.

It is also possible to consider simultaneous selection of several objects on the map, or even of a zone containing objects. In that case, content determination will be given the corresponding KB nodes and the subsequent NLG steps will calculate a subgraph of KB, and the corresponding paragraph will be generated. Nevertheless this operation raises the issue of *stability*: since textual representations of subgraphs are calculated on-the-fly, adding a node to the initial set may change the syntactic structure of the produced text completely, and this can disrupt the ENC user's attention. The solution to this problem is to keep track of consecutively selected subgraphs of KB, leave their intersection unchanged and add text only at the end.

Finally let us note that interaction between textual and visual language can work in both ways: selecting visual words can result into text production, and conversely, selecting text words can result into highlighting the corresponding visual words.

5.6.2 Collaborative updates of the knowledge base

It is important for the SHOM knowledge base to be kept constantly up-to-date. To achieve this goal, INAUT will be used as a tool for collaborative update. Indeed, INAUT has been designed as the optimal compromise between easiness of use (since contributors have a priori no knowledge management proficiency) and formality (as the knowledge base will be fed directly by the incoming data).

To make the system more robust, we validate on two levels (cf. Fig. 13). First, the lexical and syntactic level: the parser module analyzes segments written in INAUT and validates them. In case of errors it provides correction hints. Second, the semantic level: the system checks for logical consistency and informativeness.

6 Tools and development

The development of the system described in this paper is a still ongoing process involving various tools. While some parts of the system are in-house tools and algorithms, we propose here a short review of third-party tools at our disposal.

The knowledge base is stored using the NoSQL graph database Neo4j¹⁵. Unlike traditional relational databases, according to Nagi (2013), graph databases allow a certain scalability that is an advantage for the possible evolution of our system. Furthermore, Neo4j proposes a few inbuilt graph algorithms for finding paths (Dijkstra, A*, etc.) which are useful for our purposes.

To retrieve georeferenced nodes we used the query language of Neo4j, Cypher, in combination with a geodatabase and a Python library named Shapely¹⁶. Shapely is based on the PostGIS-engine GEOS and covers a large variety of geometric elements such as points, line strings, polygons, etc. The different predicates (intersects, touches, disjoint, crosses, within, contains, overlaps, equals, covers) and operations (union, distance, intersection, symmetric difference, convex hull, envelope, buffer, simplify, polygon assembly, valid, area, length) enable us to

¹⁵ <http://neo4j.com/>.

¹⁶ <https://github.com/Toblerity/Shapely>.

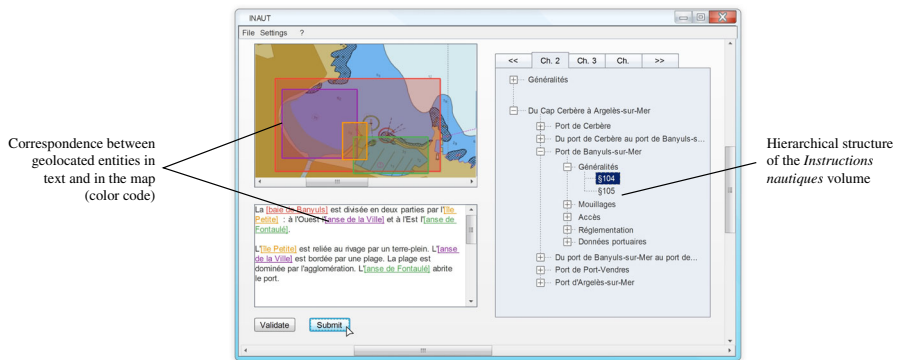


Fig. 17 Preview of the graphical user interface of the system

perform various computations to retrieve the desired areas in the KB by using Shapely functions in association with the points and polygons coordinates stored in a spatial database. It is therefore possible to retrieve the nodes corresponding to the user selection. The geodatabase is implemented within the MySQL DBMS¹⁷, although a migration of the geodatabase from MySQL to Neo4j is conceivable.

Users formulate their updates via hybrid sentences (that is: textual sentences plus a GUI to select areas in the map) which is parsed and converted into Cypher queries using an in-house CNL parser using Python libraries NLTK¹⁸ and PLY¹⁹.

The interface to the knowledge base consists of an area displaying the geographic map extract (with the possibility of drawing rectangular zones), a text entry area where the user can insert new text in INAUT (with the possibility of validating it through a dedicated button), and an area displaying the hierarchical structure of a chosen volume where the user can select the hierarchical structure tree node to which the new text belongs (see Fig. 17). After writing the text, connecting geolocated entities (written between brackets) in the text with the corresponding zones in the map, validating the textual part, and selecting the appropriate hierarchical structure node, a submit button allows to transmit the hybrid sentence to the knowledge base.

Currently this interface is written in PHP and uses the Google Maps v3 API to display maps and allow users to select points or areas on it. However the system is planned to evolve to use the NaVisu system²⁰, an open-source marine navigation software already containing a fair amount of electronic navigational charts. NaVisu is built upon the NASA WorldWind²¹ virtual globe which is geolocated and enables users to view marine charts, to place points of interest upon them, to retrieve GPS coordinates, etc.

¹⁷ <https://www.mysql.com/>.

¹⁸ <http://www.nltk.org/>.

¹⁹ <http://www.dabeaz.com/ply/>.

²⁰ <https://github.com/terre-virtuelle/navisu>.

²¹ <http://worldwind.arc.nasa.gov/java/>.

7 Evaluation

The INAUT system allows a bidirectional communication between human users and the SHOM knowledge base. The direction going from the human user to the knowledge base can hardly be evaluated since the knowledge base structure has been designed ab initio to fit with the text of the *Instructions nautiques* and therefore this operation runs smoothly and without information loss. The only kind of evaluation we can perform is user satisfaction (cf. Sect. 7.2).

The situation is quite different when going in the other direction: natural language generation algorithms generate text out of the knowledge base, and generated text must be relevant, fluid and as close as possible to legacy *Instructions nautiques* text. In this case we can measure (automatically or manually) the quality of the generated paragraphs not solely on the basis of human opinions but on the basis of rules and objective measures.

We describe in this section a pilot evaluation that takes into account the relevance and quality of generated paragraphs and the relevance of the produced document according to domain experts.

7.1 Machine-oriented evaluation

Our current setup involves 127 subsections of one *Instructions nautiques* volume covering the coastlines from the Croisette Cape to the Italian border (see Fig. 18). These subsections contain a total of 462 INAUT sentences. A total amount of 100 areas has been randomly selected on the maps and the corresponding instructions have been generated in order to create a new corpus for the evaluation.

To measure the quality of the generated *Instructions nautiques*, we use an objective measure by computing two relevant criteria: (1) the quality of the sentences ordering, and (2) the quality of the sentences splitting.

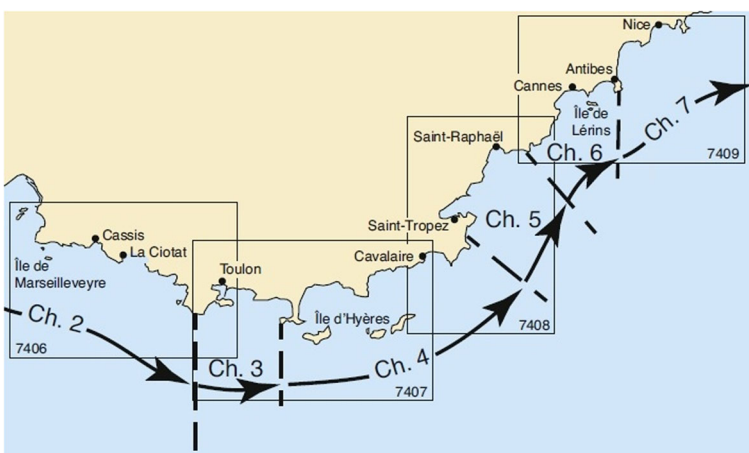


Fig. 18 From the Croisette Cape to the Italian border—the geographical area of the chosen volume of *Instructions nautiques*

For the first criterion, we compared sentence order in the legacy *Instructions nautiques* documents with sentence order obtained by the NLG discourse planning module. We used Levenshtein distance (1966) to compute differences between the original order of paragraphs and the new one. The Levenshtein distance quantifies the minimum number of insertions, deletions, and substitutions required to convert one string into another (substitution has a cost of two units while deletion and insertion have a cost of one unit).

Choosing the proper first sentence of a paragraph is an important task for the ordering process in NLG, hence we evaluated separately the choice of the first sentence. Among 100 sentences that are first sentences of legacy paragraphs: (a) 65.1 % duly occupy the first position in the generated text, (b) 18.4 % occupy the second position, (c) and 16.5 % occupy the third position or more.

Reordering the 100 paragraphs gave the following results: (a) in 47.2 % of cases the retrieved order hasn't changed, (b) in 33.2 % of cases, only one sentence has been inaccurately ordered, (c) 16.5 % of paragraphs have two disordered sentences, (d) and finally 3.1 % of paragraphs have more than two badly ordered sentences.

The second evaluated criterion takes into account the quality of the paragraphs splitting. We manually detected the ill-formed splitting²² occurrences in the 100 newly generated paragraphs. On the total amount of generated paragraphs, (a) 68.3 % of them have been properly split, (b) 22.6 % present only one split error, (c) 6.4 % have two incorrect splittings, (d) and lastly 2.7 % of paragraphs have been improperly split more than two times.

7.2 Human-oriented evaluation

The human-oriented evaluation part consisted of a small-scale evaluation by experts (mainly *Instructions nautiques* authors) and was focused on understandability and fluidity of the generated documents.

Experts were asked to rate the two criteria for the 100 generated paragraphs of the first part of the evaluation. We used this subjective measure to estimate the quality of whole paragraph generation as well as the one of document structuring subtask 4 (cf. p. 28). Experts reported very positive feedback concerning the understandability of the generated paragraphs. However they indicated more mixed opinions regarding the fluidity criterion. This can be explained by the nature of legacy *Instructions nautiques* which are often closer to literary texts than to technical texts—authors expecting the system to produce texts as fluid as their own were disappointed.

A future, more structured evaluation will include a larger group of experts and users and will allow us to obtain more precise results concerning user feedback, also involving the INAUT GUI.

²² An ill-formed splitting would be a splitting not respecting the syntax, semantics, style and legacy conventions.

8 Conclusion and future work

We have defined the notion of controlled hybrid language, and described the controlled hybrid language INAUT used for interaction with the SHOM maritime knowledge base, for automatic generation of *Instructions nautiques* documents and for interaction with ENCs.

Among our plans is the extension of INAUT into a Q&A system. This requires extension of INAUT to interrogative sentences and increased use of the concept hierarchy.

Another extension deals with the issue of *dangerousness*. Indeed, one of the goals of *Instructions nautiques* is to alert the navigator on possible dangers and risks of various kinds. Ideally, the ENC should automatically send queries about dangerousness to the knowledge base including the current position of the vessel and the meteorological/tidal context, and in case of a positive response, alert the navigator by all means possible. Special natural language generation techniques can then be used, since the communicative goal will not be simply to inform, but to alert. The hybrid nature of the language can then be used to obtain multimodal (text and image) alerts and hence increase their efficiency.

References

- Angelov, K., & Ranta, A. (2009). Implementing controlled languages in GF. In Fuchs, N. (Ed.) *Proceedings of CNL 2009, Springer lecture notes in artificial intelligence* (Vol. 5972, pp. 82–101).
- Barkmeyer, E., & Mattas, A. (2012). A restricted English for constructing ontologies (RECON). In *NIST interagency/internal report 7868*.
- Beazley, D. Python Lex-Yacc. <http://www.dabeaz.com/ply/>.
- Bertin, J. (1999). *Sémiologie graphique*. Paris: Éditions de l'EHSS.
- Camilleri, J.J., Paganelli, G., & Schneider, G. (2014). A CNL for contract-oriented diagrams. In *Proceedings of CNL 2014, LNAI* (Vol. 8625, pp. 135–146).
- Chomsky, N. (1956). Three models for the description of language. *IEEE Transactions on Information Theory*, 2, 113–124.
- Cohen, W. W., Schapire, R. E., & Singer, Y. (1999). Learning to order things. *Journal of Artificial Intelligence Research*, 10(1), 243–270.
- Costagliola, G., Ferrucci, F., Polese, G., & Vitiello, G. (1999). Supporting hybrid and hierarchical visual language definition. In *IEEE symposium on visual languages* (pp. 236–243).
- Ferrucci, F., et al. (1996). Symbol-Relation grammars: A formalism for graphical languages. *Information and Computation*, 131, 1–46.
- Freund, Y., Iyer, R. D., Schapire, R. E., & Singer, Y. (1998). An efficient boosting algorithm for combining preferences. In *Proceedings of the fifteenth international conference on machine learning, ICML'98* (pp. 170–178). San Francisco, CA: Morgan Kaufmann Publishers Inc.
- Grant Head, C. (1991). Mapping as language or semiotic system: Review and comment. In D. M. Mark & A. U. Frank (Eds.), *Cognitive and linguistic aspects of geographic space* (pp. 237–262). Berlin: Kluwer Academic Publishers.
- Haralambous, Y., Sauvage-Vincent, J., & Puentes, J. (2014). INAUT, a controlled language for the French coast pilot books instructions nautiques. In *Proceedings of CNL 2014, LNAI* (Vol. 8625, pp. 102–111).
- Ilastic. <http://www.crunchbase.com/organization/ilastic>.
- IMO. (2005). IMO standard marine communication phrases, with pronunciation. International Maritime Organization.
- International Hydrographic Association: Special publication S-57. http://www.iho.int/iho_pubs/IHO_Download.htm.

- Kerpedjiev, S.M. (1992). Automatic generation of multimodal weather reports from datasets. In *Proceedings of the third international conference on applied natural language processing* (pp. 48–55).
- Knuth, D. E. (1968). Semantics of context-free languages. *Mathematical Systems Theory*, 2, 127–145.
- Kuhn, T. (2014). A survey and classification of controlled natural languages. *Computational Linguistics*, 40, 121–170.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8), 707–710.
- Li, Z. (1995). Maps as a visual language: A Chinese perspective. Ph.D. thesis, Wilfrid Laurier University.
- Machin, D. (2007). *Introduction to multimodal analysis*. London: Bloomsbury Academic.
- Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8(3), 243–281.
- Marriott, K., Meyer, B., & Wittenburg, K.B. (1998). A survey of visual language specification and recognition. In: *Visual language theory* (pp. 5–85). Springer.
- Menanteau, B. (Ed.) (2011). Édition des Instructions nautiques (procédure spécifique). SHOM.
- Menanteau, B. (Ed.) (2013). Définition des Instructions nautiques du SHOM (norme). SHOM.
- Morvan, S., Marques, D., Mens, J., Pensec, T., & Tritschler, C. NaVisu project. <https://github.com/terre-virtuelle/navisu>.
- MySQL AB: MySQL. <https://www.mysql.com/>.
- Nagi, K. (2013). A new representation of WordNet using graph databases. *International Journal on Advances in Software*, 6, 298–308.
- NASA: WorldWind API. <http://worldwind.arc.nasa.gov/java/>.
- Natural Language Toolkit. <http://www.nltk.org/>.
- Nečas, I. (2011). BDD as a specification and QA instrument. Master's thesis, Masaryk University.
- Neo4j project. <http://neo4j.com/>.
- NOAA: United States coast pilot. <http://www.nauticalcharts.noaa.gov/nsd/cpdownload.htm>.
- Pool, J. (2006). Can controlled languages scale to the Web? In *Proceedings of the 5th international workshop on controlled language applications* (pp. 1–12).
- Pratt-Hartmann, I. (2013). Twenty years of topological logic. In M. Raubal, D. M. Mark, & A. U. Frank (Eds.), *Cognitive and linguistic aspects of geographic space, lecture notes in geoinformation and cartography* (pp. 217–236). Berlin: Springer.
- Randell, D.A., Cui, Z., & Cohn, A.G. (1992). A spatial logic based on regions and connection. In *Proceedings of the 2nd international conference on knowledge representation and reasoning* (pp. 165–176). Morgan Kaufmann.
- Ranta, A. (2009). Grammatical framework: A multilingual grammar formalism. *Language and Linguistics Compass*, 3, 1242–1265.
- Reiter, E., & Dale, R. (2000). *Building natural language generation systems*. Cambridge: Cambridge University Press.
- Roth, M., & Frank, A. (2010). Computing EM-based alignments of routes and route directions as a basis for natural language generation. In *Proceedings of the 23rd international conference on computational linguistics*, (pp. 958–966). Association for Computational Linguistics.
- Sauvage-Vincent, J., Haralambous, Y., & Puentes, J. (2015). Sentence ordering in electronic navigational chart companion text generation. In *ENLG 2015: 15th European Workshop on natural language generation* (pp. 66–70). Stroudsburg, PA: The Association for Computational Linguistics.
- Schlichtmann, H. (1984). Characteristic traits of the semiotic system “map symbolism”. *The Cartographic Journal*, 22, 23–30.
- Schwiter, R. (2002). English as a formal specification language. In: *Proceedings of DEXA '02* (pp. 228–232).
- Shapely Python package. <https://github.com/Toblerity/Shapely>.
- SHOM: Instructions nautiques. <http://www.shom.fr/les-produits/produits-nautiques/ouvrages-nautiques/instructions-nautiques/>.
- SHOM. (2012). Symboles, abréviations et termes utilisés sur les cartes marines, Vol. INT 1, 5th edn.
- Strevens, P., & Johnson, E. (1983). SEASPEAK: A project in applied linguistics, language engineering, and eventually ESP for sailors. *The ESP Journal*, 2(2), 123–129.
- UKHO: Admiralty sailing directions. <https://www.ukho.gov.uk/PRODUCTSANDSERVICES/PAPER/PUBLICATIONS/Pages/NauticalPubs.aspx>.

- Varanka, D. (1991). An approach to map/text interrelationships. In D. M. Mark & A. U. Frank (Eds.), *Cognitive and linguistic aspects of geographic space* (pp. 285–294). Berlin: Kluwer Academic Publishers.
- Walker, M.A., Rambow, O., & Rogati, M. (2001). Spot: A trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies* (pp. 1–8). Association for Computational Linguistics.