The aim of this presentation is to argue that a specific computational component is needed in order to manage the distribution of various portions of the signing space in sign language (SL) production. This component is specific for SLs in the sense that it is required by the visuo-gestural modality and it characterizes as a condition imposed at the articulatory-perceptive interface.

In standard models for language generation, like the Y model in the mainstream generative tradition [1], the faculty of language is represented as a modular system composed by a computational system, and two interpretative modules: the conceptual-intentional (CI) system and the articulatory-perceptive (AP) system [3]. Under a phase-theory approach [2], sentence-building proceeds cyclically through recursive application of the merge operation in the computational system (i.e. the core syntactic component). At the end of each phase (i.e. at Spell-out), the hierarchical structure generated by this mechanism is sent to the interpretative modules via the AP and CI interfaces. At the AP-interface several operations applies to the output of the syntactic component in order to produce the input for the phonological component. The most important ones are the linearization of the hierarchical structure, the operations connected with the morphological component and the lexical insertion. The linearization process takes the hierarchical structure generated at the end of a phase as an input and produces a linearly ordered sequence of items [4]. Concretely, this is shown by the example in (1), where the target sentence is that of a typical head-final (SOV) language.

(1) a. \{C \{ v, \{\{SIGN, \{CONTRACT\}\}, \{GIANNI\}\}\} T\} }
   \{GIANNI\} CONTRACT SIGN v T C

   ‘Gianni signed the contract’

This would suffice for the derivation to continue in spoken languages. However, in the case of SLs the linear order is not a sufficient output for the derivation to go on. On top of having a linear sequence of signs, there must be some further information on where hands have to be positioned in the signing space (at least for those signs that are not body anchored). Namely, signs that are produced in neutral space are not randomly distributed in the signing space, nor they are rigidly distributed right-to-left (or left-to-right).

My proposal is that there is a specific computational process that produces the correct order information for hands’ position in addition to the linearization algorithm and that this process is linguistically driven (i.e. it is part of the faculty of language in the broad sense). In a nutshell, the task accomplished by this process is similar to that of the linearization algorithm in that it receives a hierarchical structure as an input and uses syntactic information to map syntactic objects onto default spatial positions (unless other constraints are operative). Consider for concreteness, verb arguments in Italian sign language (LIS); their default positions are defined as in (2):

(2) a. Subjects are mapped onto the ipsilateral position within the relevant portion of the space
   b. Objects are mapped onto the controlateral position within the relevant portion of the space

Let’s take the pair in (3) as an illustrative example. In the case of left-handed signers, the subject GIANNI is produced in the left part of the signing space (ipsilateral), while the object MOUSE is on the right side (controlateral). This is illustrated by the pictures in (4), below.

(3) a. GIANNI MOUSE SAVE
   ‘Gianni saved the mouse’
   b. MOUSE GIANNI SAVE
   ‘Gianni saved the mouse’

Crucially, the spatial positions are not affected by the different linear orders of the examples in (3a-b), proving that a mere functionalist explanation is inadequate (i.e. it’s not the case the first element is mapped onto the ipsilateral position for ease of articulation reasons.). Interestingly enough, the spatial position is computed even when arguments are not articulated in neutral space. Namely, even when subjects are body-anchored signs, objects are articulated in the controlateral position.
The spatial algorithm applies cyclically at least at end of every CP phase. This is illustrated by the case of sentential objects constructions in (5). The sentential object (PIERO BIKE FELL) is produced in the contralateral portion of the space (as predicted by 2b) and in that position the space is further splitted into a more ipsilateral position for the subordinate subject PIERO (as predicted by 2a) and a more contralateral position for the object BIKE (as predicted by 2b). The matrix subject GIANNI is articulated in the standard ipsilateral position (as predicted by 2a).

(5) \[\text{CP [CP PIERO BIKE FELL] GIANNI TELL} \] ‘Gianni told that Piero fell from the bike’

In a similar fashion, sentential subjects are produced in the ipsilateral position of the signing space. More interesting is the case of raising predicates, where the two linear sequences in (6) are available (roughly corresponding to the raised vs. non-raised subject construction).

(6) a. SEEM [GIANNI MOUSE SAVE] ‘it seems that Gianni saved the mouse’
    b. GIANNI SEEM MOUSE SAVE ‘Gianni seems to have saved the mouse’

In (6a), where presumably a null expletive is employed, the entire sentential complement of SEEM is produced in the contralateral position, while in (6b) the raised subject is in the ipsilateral position (as expected) and the rest of the complement is in the contralateral position. This optionality does not seem to be available in the case of control constructions like (7) where only the spatial distribution of (6b) is attested.

(7) GIANNI [MILK BUY ] FORGET ‘Gianni forgot to buy the Milk’

I argue that the spatial distribution in (6b) and (7) is the instantiation of a more general effect due to argument sharing. Although in different syntactic ways, both in (6b) and in (7) the matrix subject is shared with the sentential complement. Along this line, I will also analyze relative constructions. Finally, I will discuss cases where the effects of the spatial algorithm are blurred by phonological constraints imposed by specific lexical items (e.g. two-handed signs tend to be centered) and evaluate them against the general architecture of language.

To conclude, current models for language generation based solely on the grammar of spoken languages are not adequate to capture all the computational processes needed to generate grammatical sentences in SLs. My proposal is to revise the standard Y model by introducing an additional computational step that manages the appropriate allocation of spatial resources at the AP-interface. This immediately raises non-trivial questions about the cognitive effort in terms of planning and working-memory load. I will address some of them in the concluding remarks.

References