Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 324 (2016) 51–66

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Elements for a Formal Model of Intentional Systems

Antônio Carlos da Rocha Costa[1,2]

Programa de Pós-Graduação em Informática na Educação - PGIE
Universidade Federal do Rio Grande do Sul - UFRGS
Porto Alegre, RS, Brazil

Programa de Pós-Graduação em Computação - PPGComp
Universidade Federal do Rio Grande - FURG
Rio Grande, RS, Brazil

Abstract

This paper introduces elementary concepts needed to define a formal, computationally oriented, model for *intentional systems*. First, the paper briefly reviews the central concept of *intentionality*, to contextualize the work. Then, it characterizes the main types of *intentional acts*, defines the concepts of *intentional process* and *intentional system*, and gives the basis of the formal model of such systems. Next in a brief case study, a formal model for a sample *constative* intentional system is presented and discussed. Following, the features that are still lacking to achieve a full-fledged formal model of intentional systems are indicated. Finally, the relationship between the formal model of intentional systems introduced here and the usual semantical models for formal languages is discussed.

Keywords: Phenomenology, intentionality, intentional processes, intentional systems, formal semantics.

1 Introduction

Phenomenology [10] is the area of Philosophy that studies the structure of *experience* and the *acts of consciousness*, reflexive or not, that constitute it.

The characteristic feature of an act of consciousness is its *intentionality*, that is, its directedness toward an object. Acts of consciousness are, thus, *intentional acts*. Temporal sequences of intentional acts constitute what may be called *intentional processes*. We call *intentional system* any system that performs an intentional process.

In this paper, we introduce elements for a formal model of intentional systems which is computationally oriented, meaning that it is conceived having its computa-

[1] CNPq provided partial financial support for this work.

[2] Email: `ac.rocha.costa@gmail.com`

http://dx.doi.org/10.1016/j.entcs.2016.09.006
1571-0661/

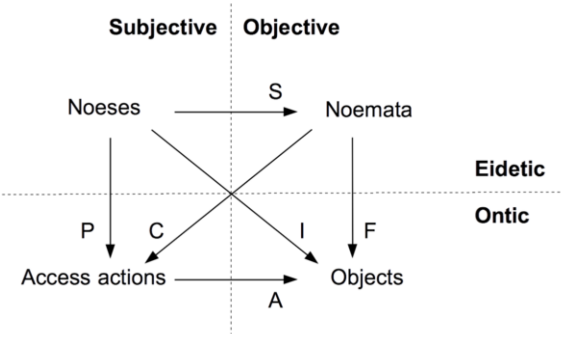

Fig. 1. The Husserlian square.

tional realization, through agent technologies, in multiagent systems [11]. We base the work on Edmund Husserl's analysis of intentional acts [6].

The paper is structured as follows. In Sect. 2, we review the concepts of *intentional act* and *intentional process*, the main background concepts of the work. In Sect. 3 we introduce the formal concept of *intentional system*, on the basis of which we realize the formal modeling of intentional processes. Sect. 4 presents a simple case study, giving a concrete example of the applicability of the ideas presented here. Sect. 5 is the Conclusion, where some related works and some general issues are discussed.

2 Intentional Acts and Intentional Processes

2.1 Husserl's Concept of Intentional Act

The way we construe in formal, computationally oriented, terms Husserl's analysis of intentional acts [6] is illustrated in Fig. 1, where we picture, in what we call the *Husserlian square*, four constituents of intentional acts. They are[3]:

- *Objects*: the set of things and facts in the world toward which intentional acts may be directed; formally denoted by **Objs**.
- *Access actions*: the effective actions[4] through which the objects may be accessed, as such actions occur in the *intentional systems* that realize the intentional acts; formally denoted by **AActs**.
- *Noeses*: the set of ideated essences of access actions, as each such essence (*noesis*) is grasped phenomenologically[5]; formally denoted by **Noes**.
- *Noemata*: the set of ideated essences of objects and facts, as each such essence (*noema*) is grasped phenomenologically; formally denoted by **Nmts**.

The relationships between the various constituents of the intentional acts are:

[3] Note how the Husserlian square subsumes, through its *subjective* × *objective*, and *eidetic* × *ontic* categorization, the distinction between the *epistemic* and the *ontological* senses of the *subjective* × *objective* distinction, extensively elaborated by Searle in, e.g., [9].

[4] "Effective" in the computational sense, i.e., endowed with only finitary features, restricting their applicability to (parts of) objects that can be accessed in finite time.

[5] The *phenomenological method* of grasping objects and mental actions is *observational*, aiming at the *description* of the manifest aspects of the phenomena of consciousness that it studies [6].

- Functional relationships:
  · $\mathrm{A} : \mathbf{AActs} \to \mathbf{Objs}$ is the *access* function;
  · $\mathrm{S} : \mathbf{Noes} \to \mathbf{Nmts}$ is the *sense* function;
  such that:
  · if $\mathrm{A}(a) = o$, we say that access action a can access object o, when executed;
  · if $\mathrm{S}(ns) = nm$, we say that the noema nm is the sense of the noesis ns.
- Relational relationships:
  · $\mathrm{P} \subseteq \mathbf{Noes} \times \mathbf{AActs}$ is the *performance* relation;
  · $\mathrm{F} \subseteq \mathbf{Nmts} \times \mathbf{Objs}$ is the *fulfillment* relation;
  · $\mathrm{I} \subseteq \mathbf{Noes} \times \mathbf{Objs}$ is the *intentional* relation;
  · $\mathrm{C} \subseteq \mathbf{Nmts} \times \mathbf{AActs}$ is the *compatibility* relation;
  such that:
  · if $a \in \mathrm{P}[ns]$, we say that the access action a performs the noesis ns, when executed;
  · if $o \in \mathrm{F}[nm]$, we say that the object o fulfills the noema nm;
  · if $o \in \mathrm{I}[ns]$, we say that the object o is intended by the noesis ns;
  · if $a \in \mathrm{C}[nm]$, we say that the action a is compatible with the noema nm.

If an act performed by an intentional system is *intentional* then its structure fits the structure of the Husserlian square. That is, we define, in a preliminary way [6] :

Definition 2.1 An *intentional act* is a tuple of the form

$$(ns, aact, nt, obj) \in \mathbf{Noes} \times \mathbf{AActs} \times \mathbf{Nmts} \times \mathbf{Objs}$$

The universe of all possible intentional acts is denoted by **IActs**.

2.2 Intentional Processes

By a (general) *process* we understand any time-indexed sequence of sets of actions. We say that a process is *intentional* if at each time instant at least one of the actions that occur in the process, at that time, is an *intentional act*.

An *intentional process* is said to be *pure* if all its sets of actions are constituted only by intentional acts.

For simplicity, we consider here intentional processes constituted only by non-empty sets with just one single act, which happens to be intentional, that is, an element of **IActs**. And, we take time to be discrete and linearly ordered, denoted by $T = 0, 1, 2, \dots$.

Under such conditions, we may define, in a reductive way:

Definition 2.2 An *intentional process* is any time-indexed sequence $ip : T \to \mathbf{IActs}$ of intentional acts.

With the set of intentional acts occurring at time t denoted by ip^t, for $t = 0, 1, 2, \dots$, we may write $ip = \langle ip^0, ip^1, ip^2, \dots \rangle$. Each ip^t is said to be a *step* of ip. We denote the universe of all possible intentional processes by **IProc**.

---

[6] The concept of intentional act is further refined in Sect. 2.3.

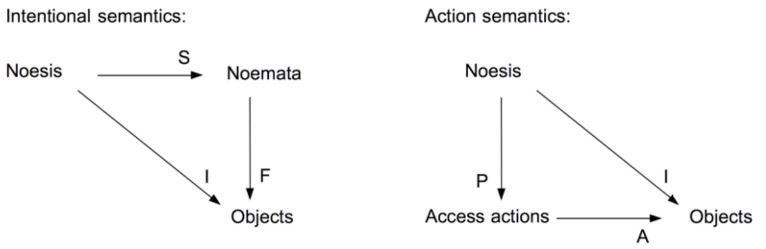

Fig. 2. The Husserlian triangles of the intentional and actional semantics of intentional acts.

2.3 Formal Semantics of Intentional Acts

We determine the formal semantics of intentional acts through a combination of two types of semantical functions, the *intentional semantical function* and the *actional semantical function*:

Definition 2.3 An *intentional semantical function* for intentional acts is any function $\mathrm{ISem} : \mathbf{IActs} \to \mathbf{Noes} \times \mathbf{Nmts} \times \wp(\mathbf{Objs})$, such that if, for any $ia \in \mathbf{IActs}$, one has $\mathrm{ISem}(ia) = (ns, nm, O)$ then it holds that:

- $nm = \mathrm{S}(ns)$, i.e., nm is the sense of ns;
- $O = \mathrm{F}(nm)$, i.e., O is the set of objects that fulfill nm;
- $O = \mathrm{I}(ns)$, i.e., O is the set of objects intended by ns;

and $\mathrm{I} = \mathrm{F} \circ \mathrm{S}$, so that the upper right triangle in Fig. 1 commutes.

Definition 2.4 An *actional semantical function* for intentional acts is any function $\mathrm{ASem} : \mathbf{IActs} \to \mathbf{Noes} \times \wp(\mathbf{AActs}) \times \wp(\mathbf{Objs})$, such that if, for any $ia \in \mathbf{IActs}$, one has $\mathrm{ASem}(ia) = (ns, A, O)$, then it holds that:

- $A = \mathrm{P}(ns)$, i.e., A is the set of actions that can perform ns;
- $O = \mathrm{A}(A)$, i.e., O is the set of objects accessed by the actions of A;
- $O = \mathrm{I}(ns)$, i.e., O is the set of objects intended by ns;

and $\mathrm{I} = \mathrm{A} \circ \mathrm{P}$, so that the lower left triangle in Fig. 1 commutes.

The two types of semantical functions of intentional acts are shown in Fig. 2, as *Husserlian triangles.*

We define a notion of compatibility between the intentional and the actional semantical functions:

Definition 2.5 Given an intentional semantical function $\mathrm{ISem} : \mathbf{IActs} \to \mathbf{Noes} \times \mathbf{Nmts} \times \wp(\mathbf{Objs})$ and an actional semantical function $\mathrm{ASem} : \mathbf{IActs} \to \mathbf{Noes} \times \wp(\mathbf{AActs}) \times \wp(\mathbf{Objs})$, these two semantical functions are said to *compatible with each other* if and only if, for any intentional act $ia \in \mathbf{IActs}$, whenever $\mathrm{ISem}(ia) = (ns, nm, O)$ and $\mathrm{ASem}(ia) = (ns', A, O)$ it holds that: $ns = ns'$ and $A = \mathrm{C}(nm)$.

Two central concepts that we define are the following. First, the concept of well-foundedness of a Husserilan square:

Definition 2.6 A formally determined Husserlian square is *well-founded* only when its intentional semantical function ISem : $\mathbf{IActs} \to \mathbf{Noes} \times \mathbf{Nmts} \times \wp(\mathbf{Objs})$ and its actional semantical function ASem : $\mathbf{IActs} \to \mathbf{Noes} \times \wp(\mathbf{AActs}) \times \wp(\mathbf{Objs})$ are compatible with each other.

Second, the concept of an intentional act being well-defined, relatively to a pair of intentional and actional semantical functions:

Definition 2.7 Given the semantical functions ISem : $\mathbf{IActs} \to \mathbf{Noes} \times \mathbf{Nmts} \times \wp(\mathbf{Objs})$ and ASem : $\mathbf{IActs} \to \mathbf{Noes} \times \wp(\mathbf{AActs}) \times \wp(\mathbf{Objs})$, compatible with each other, we say that an intentional act $ia = (ns, aact, nm, obj) \in \mathbf{IActs}$ is *well defined*, relatively to ISem and ASem, if and only if, whenever $\text{ISem}(ia) = (ns, nm, O)$ and $\text{ASem}(ia) = (ns, A, O)$, it holds that: $aact \in A$ and $obj \in O$.

In the following, we restrict ourselves to well-defined intentional acts of well-founded Husserlian squares.

3 Intentional Systems and the Temporal Evolution of Intentional Processes

We define now intentional systems and their temporal evolutions:

Definition 3.1 An *intentional system* is a structure $IS = (ip, ISR, ASR)$, where[7]:

- $ip : T \to \mathbf{IAct}$ is the intentional process performed by IS;
- $ISR \in \wp(\mathbf{ISR})$ is a set of *intentional semantical rules*, which define the *intentional semantical function* of the system;
- $ASR \in \wp(\mathbf{ASR})$ is a set of *actional semantical rules*, which define the *actional semantical function* of the system.

The temporal evolution of an intentional system, determined by its intentional process, is given by:

Definition 3.2 Given the intentional system $IS = (ip, ISR, ASR)$, the *temporal evolution* of IS, determined by ip, is a sequence of *intentional transitions*:

$$ic^0 \xrightarrow{R^0} ic^1 \xrightarrow{R^1} ic^2 \ldots$$

where, for any time t:

- the stream $ic^t = ip^t.ic^{t+1} = (ns^t, act^t, nm^t, o^t).ic^{t+1}$ is said to be the *intentional configuration* of the intentional system IS at the time t;
- for each configuration transition $ic^t \xrightarrow{R^t} ic^{t+1}$, it happens that R^t is the *rule of configuration transition* (taken from $ISR \cup ASR$) that is applied to the configuration ic^t.

[7] **ISR** is the set of all possible *intentional semantical rules*, **ASR** is the set of all possible *actional semantical rules*.

In the following, we restrict the intentional reactive processes to encompass just two types of intentional acts:

- reactive intentional acts capable of ideating input actions, realized in the environment, as noeses of the form $\downarrow(nm)$, which substitute the noema nm for the current noema, whatever it is, in the current configuration of the intentional system;
- reactive intentional acts capable of ideating input actions, realized in the environment, as noeses of the form $nm{\downarrow}nm'$, which substitute the noema nm' for the current noema nm in configuration of the intentional system, with nm' possibly being a function of nm.

We formally present in Fig. 3, in the *natural deduction style* introduced by Plotkin [8], the intentional and actions semantical rules of the sample intentional system we are considering.

To take into account the input and output actions of the reactive intentional system, we have extended the structure of the configurations, in the rules of Fig. 3, with the respective input and output processes, $inp = \langle inp^0, inp^1, \ldots\rangle$ and $out = \langle out^0, out^1, \ldots\rangle$, so that: $ic^t = (inp^t, [ns^t, act^t, nm^t, o^t], out^t)$.

Figure 4 illustrates the general aspect of the sequencing of reactive intentional acts, in the temporal evolutions of the given reactive intentional system (with input action enforcing the application of rule ASR_{inp_1}). Notice how the sequencing organizes the temporal evolution in *reactive cycles*.

We remark the following about the semantical rules in Fig. 3:

- only the rules ASR_{inp_1} and ASR_{inp_1} can start a cycle; the cycles start with configurations of the form $(inp, [\perp, \perp, \perp, \perp], out)$;
- the intentional semantical rules take the compatibility conditions into account;
- the symbol $\perp$ denotes an undefined element;
- $i.inp'$ denotes that inp has next input i and continuation inp'; dually, $out'.o$ denotes that out' is the sequence of outputs produced before the output o;
- PI denotes the input operation and the ideation to which it gives rise;
- rule ASR_{inp_2} can only happen by interrupting an ongoing reaction;
- in the rule ASR_{out}, the configuration $(inp, \perp, \perp, \perp, \perp, out.aact(obj))$ marks the successful completion of a reaction to a perception, the output of $aact$ being indicated by $aact(obj)$.
- $t + 3$ is the earliest time at which a new input action can be realized without interrupting the reactive cycle initiated by at time t.

Regarding the first remark, notice that situations in which an input action is available to be performed, and the current configuration allows for the realization of an output action, are the only situations where indeterminacy may arise. And that, only between the realization of the input action (through rules ASR_{inp_1} or ASR_{inp_2}), and the realization of an output action (through rule ASR_{out}). For the rest, the temporal evolutions generated by the given transition rules are deterministic.

Actional Semantical Rules:

$$\frac{\mathrm{PI}[i] = \downarrow(nm)}{(i.inp, [ns, aact, nm, obj], out) \to (inp, [\downarrow(nm), \perp, \perp, \perp], out)}\ ASR_{inp_1}$$

$$\frac{\mathrm{PI}[i] = (nm)\downarrow(nm')}{(i.inp, [\downarrow(nm), aact, nm, obj], out) \to (inp, [\downarrow(nm'), aact, nm, obj], out)}\ ASR_{inp_2}$$

$$\frac{}{(inp, [ns, aact, nm, obj], out) \to (inp, [\perp, \perp, \perp, \perp], out.aact(o))}\ ASR_{out}$$

Intentional Semantical Rule:

$$\frac{aact \in \mathrm{P}[\downarrow(nm)] \quad aact \in \mathrm{C}[nm] \quad obj \in \mathrm{F}[nm] \quad obj \in \mathrm{I}[\downarrow(nm)]}{(inp, [\downarrow(nm), \perp, \perp, \perp], out) \to (inp, [\downarrow(nm), aact, nm, obj], out)}\ ISR_{\downarrow(nm)}$$

Fig. 3. The intentional and actional semantical rules of the sample formal semantics of intentional processes.

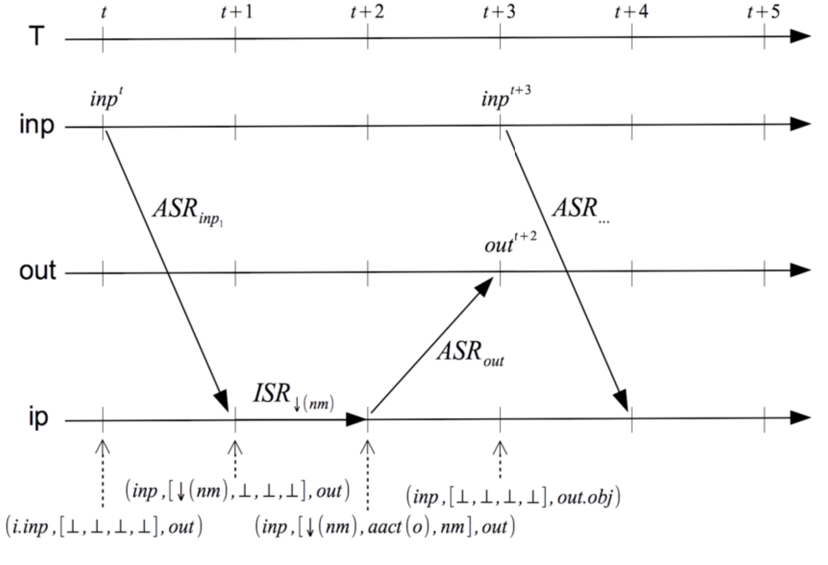

Fig. 4. A reactive intentional cycle of the temporal evolution generated by the rules of Fig. 3.

4 Case Study: A Constative Intentional System and its Formal Model

4.1 Judgments and Constative Intentional Processes

We take a *judgement* (or *constative intentional act* [8]) to be an intentional act whose noema is a *proposition*, and we take a proposition to be an *idea that a predication is true of an object of a given type.*

A (pure) *constative intentional process* is an intentional process whose intentional actions are all constative, that is, are all *judgements*. A constative intentional process is said to be *impure* if it contains other types of acts, besides intentional ones.

That a predication is true of an object constitutes what can be called a *fact*. Thus, we may say that the noema of a constative intentional act, which is a proposition, when it is fulfilled, it is fulfilled by a fact.

In a strict sense, a fact fulfills a proposition if and only if (i) the fact is constituted by a predication that is true of an object, (ii) the object and the predication are those specified by the judgment [9].

In a more general sense, one admits that facts may fulfill propositions only in a partial way (as is the case of any object, regarding a noema). The fulfillment of a proposition by a fact is *partial* if (i) either the object (or the predication, or both) only partially fulfills the condition of being of the type specified by the proposition and/or (ii) the predication is true of the object only in a partial way.

4.2 The Informal Presentation of the Case Study

Let's consider a software agent ag that is responsible for checking the completeness of a repository of learning objects LOR against a list of requirements. Fig. 5 illustrates the situation:

- ag is the *agent*;
- LOR is the *repository of learning objects*;
- pct is the operation of *perception*, through which ag accesses LOR;
- id is the operation of *ideation*, through which ag constructs Husserlian squares for the objects in LOR;
- the tuples of the form $(ns, aact, nm, obj)$ constitute the Husserlian squares, one for each checking of objects in LOR.

Let's assume that the checking should proceed on the basis of the agent ag receiving periodically, from another agent ag', a request to check if the repository LOR satisfies some specific requirement, and let's consider only the following types

[8] Clearly, we took the term "constative" from Austin [1]

[9] We keep undetermined here the detailed specification of when one may say that a predication is true of an object and when one may say that an object (or a predication) fulfills the condition of being of a given type. See discussion in Sect. 5, regarding the latter point.

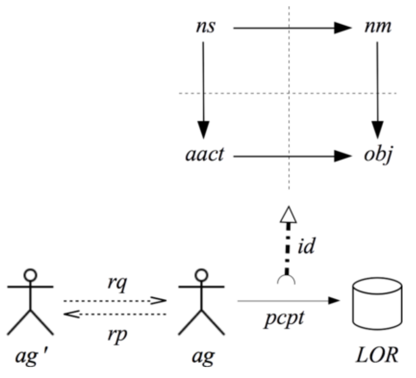

Fig. 5. Case study: agent ag checking the repository of learning objects LOR.

of requirements: (i) minimum number of learning objects, (ii) presence of a specific learning object, and (iii) maximum number of versions of a specific learning object.

We denote by rq the sequence of *requests* that ag receives from ag', and by rp the *replies* that ag gives to ag', one for each request it receives.

We take rq to be $rq = \langle rq^0, rq^1, \ldots \rangle$ where each request rq^t is of one of the types:

- $check(num\text{-}los \geq n)$: check if there are at least n learning objects in LOR;
- $check(exists(lo))$: check if the learning object lo is in LOR;
- $check(num\text{-}vers(lo) \leq n))$: check if the number of versions of the learning object lo is less than or equal to n.

with rp correspondingly being $rp = rp^0, rp^1, \ldots$ where each reply rp^t is one of the values, *True* or *False*.

4.3 An Intentional Semantical Function for the Constative Intentional Acts

Table 1 shows one possible intentional semantical function for the constative intentional acts of the agent ag.

Notice that, for the sake of space, we have not indicated the full *objects* of the constative intentional acts (which are *facts*, since the noemata of such acts are *propositions*), but just the *thing* (or pair of *things*) to which the noemata refer.

Notice, however, that such *things* are not the things that exist in LOR, but are[10] the things that the *perception* operation is capable of handling to ag, and that ag is capable of *ideating*.

These *perceived things* do not necessarily coincide completely and faithfully with what really is in LOR. That's why such things are marked with the prefix *p-* in the table.

On the other hand, notice that we are assuming, in this case study, that every (perceived) object completely fulfills its respective noema.

Finally, notice that, since the requests from ag' are supposed to be independent of each other, every new request that arrives causes the creation of a new noema

[10] As Husserl repeatedly emphasized [6].

Type of Constative Act	Noesis	Noema	Things
$check(num\text{-}los \geq n)$	$\downarrow(num\text{-}los\,(p\text{-}LOR)) \geq n)$	$num\text{-}los\,(p\text{-}LOR) \geq n$	$p\text{-}LOR$
$check(exists(lo))$	$\downarrow(exists(p\text{-}lo, p\text{-}LOR)))$	$exists(p\text{-}lo, p\text{-}LOR)$	$(p\text{-}lo, p\text{-}LOR))$
$check(num\text{-}vers(lo) \leq n)$	$\downarrow(num\text{-}vers(p\text{-}lo, p\text{-}LOR) \leq n)$	$num\text{-}vers(p\text{-}lo, p\text{-}LOR) \leq n$	$(p\text{-}lo, p\text{-}LOR)$

Table 1
The intentional semantical function for the constative intentional acts of ag.

Type of Constative Act	Noesis	Access Actions	Things
$check(num\text{-}los \geq n)$	$\downarrow(num\text{-}los\,(p\text{-}LOR)) \geq n)$	`m<-number-los(p-LOR)); return eval(m >= n)`	$p\text{-}LOR$
$check(exists(lo))$	$\downarrow(exists(p\text{-}lo, p\text{-}LOR)))$	`return eval(p-lo in p-LOR)`	$(p\text{-}lo, p\text{-}LOR))$
$check(num\text{-}vers(lo) \leq n)$	$\downarrow(num\text{-}vers(p\text{-}lo) \leq n)$	`m <- num-vers(p-lo,p-LOR); return eval(m <= n)`	$(p\text{-}lo, p\text{-}LOR)$

Table 2
The actional semantical function for the constative intentional acts of ag.

in the intentional process performed by ag. That's why all the noeses are of the $\downarrow(nm)$ type.

4.4 An Actional Semantical Function for the Constative Intentional Acts

Given any intentional semantical function *isem* for the constative intentional acts of ag, any proposed actional semantical function *asem* for those acts can be tested for its compatibility with *isem*.

Table 2 gives an actional semantical function for the constative intentional acts of ag that is compatible with the intentional semantical function given in Table 1.

In the present case study, there is a *natural notion of compatibility* between the two semantical functions. Since we are expressing the access actions of the constative intentional acts in algorithmic terms, and given that the noemata of the constative intentional acts are propositions, the natural notion of compatibility is simply that of taking the noemata as *correctness conditions* for the algorithms that represent the access actions.

For instance, concerning the constative intentional action $check(num\text{-}los \geq n)$ one has only to prove the truth of the correctness condition:

`pre:` $True$

{`m<-number-los(p-LOR));return eval(m >= n)`}

`post:` $num\text{-}los(p\text{-}LOR) \geq n$

That the algorithm terminates, it is immediately sure.

4.5 The Temporal Evolution of the Constative Intentional Process

As mentioned before, the requests sent to the checking agent ag are supposed to be independent of each other. Thus, the noesis of every intentional act in the intentional process of requests is of the form $\downarrow(nm)$.

In such condition, the temporal evolution of the intentional process consists, simply, in the temporal succession of sequences of steps like that shown in Fig. 4, since no intentional act builds on the previous one.

Since all the requests in rq have the form $\downarrow nm$, the only rules that are applicable to the intentional process are the rules ASR_{inp_1}, ASR_{out} and $ISR_{\downarrow(nm)}$ (see Fig. 3).

Thus, if:

$$rq = \langle check(num\text{-}los \geq 0), check(exists(lo_1)), check(num\text{-}vers(lo_1) \leq 10), \ldots\rangle$$

is the beginning of a sequence of requests, and the learning object repository that ag perceives, $p\text{-}LOR$, is such that:

- there are learning objects in $p\text{-}LOR$;
- the learning object $p\text{-}lo_1$ exists in $p\text{-}LOR$;
- there are more then 10 versions of the learning object $p\text{-}lo_1$ in $p\text{-}LOR$;

the temporal evolution of the intentional process performed by ag, in the given intentional system, has the initial part shown in Fig. 6, where [11] :

- ε denotes the empty sequence;
- at each step, we have omitted the continuation of the input sequence as well the beginning of the output sequence (indicated by the mark "$\sim$");
- the initial configuration is completely undetermined, except for the input sequence;
- comments are inserted within braces.

5 Discussion, Related Works, and Conclusion

5.1 Husserlian Semantics and its Alternatives

We submit here, building on the exposition above, that one advantage of adopting an intentional approach to the modeling of computational systems, specially if in accordance with Husserl's prespective [6], is the completeness of the treatment of the semantical issues of the languages involved in such types of systems.

To see that, one can directly compare the scope of semantical issues tackled by the perspecive introduced here, with the scope of semantical issues tackled by the usual semantical perspectives adopted in formal language systems (for both logical language systems, see e.g. [2], and computational language systems, see e.g. [3]).

In Fig. 8, we show the differences, regarding the semantical scopes, between the Husserlian semantical perspective, the *logical* perspective (which encompasses most of the semantics of computational languages) and the *empiricist* perspectives. The latter perspectives are the most common alternatives to the Husserlian one.

Checking against Fig. 1, one sees that logical semantics suppresses *subjectivity* from language, while empiricist semantics suppresses *subjectivity* and *eideticity*.

[11] Compare with the generic temporal structure of the process cycles, shown in Fig. 4.

$$
\begin{array}{l}
\{\text{initial configuration:}\}\\[4pt]
(\mathit{check}(\mathit{num\text{-}los} \geq 0).\mathit{check}(\mathit{exists}(lo_1)).\mathit{check}(\mathit{num\text{-}vers}(lo_1) \leq 10).\sim,\\
\quad [\bot, \bot, \bot, \bot],\\
\quad \varepsilon)\\[6pt]
\xrightarrow{ASR_{inp\,1}} \{\text{ideate next input}\}\\[4pt]
(\mathit{check}(\mathit{exists}(lo_1)).\mathit{check}(\mathit{num\text{-}vers}(lo_1) \leq 10).\sim,\\
\quad [\downarrow(\mathit{num\text{-}los}\,(p\text{-}LOR)) \geq 0), \bot, \bot, \bot],\\
\quad \varepsilon)\\[6pt]
\xrightarrow{ISR_{\downarrow(nm)}} \{\text{determine actionality and intentionality}\}\\[4pt]
(\mathit{check}(\mathit{exists}(lo_1)).\mathit{check}(\mathit{num\text{-}vers}(lo_1) \leq 10).\sim,\\
\quad [\downarrow(\mathit{num\text{-}los}\,(p\text{-}LOR)) \geq 0), \begin{array}{l}\texttt{m<-number-los(p-LOR));}\\ \texttt{return eval(m >= 0)}\end{array}, \mathit{num\text{-}los}\,(p\text{-}LOR) \geq 0, p\text{-}LOR],\\
\quad \varepsilon)\\[6pt]
\xrightarrow{ASR_{out}} \{\text{complete the access action}\}\\[4pt]
(\mathit{check}(\mathit{exists}(lo_1)).\mathit{check}(\mathit{num\text{-}vers}(lo_1) \leq 10).\sim,\\
\quad [\bot, \bot, \bot, \bot],\\
\quad \mathit{True})\\[6pt]
\xrightarrow{ASR_{inp\,1}} \{\text{ideate next input}\}\\[4pt]
(\mathit{check}(\mathit{num\text{-}vers}(lo_1) \leq 10).\sim,\\
\quad [\downarrow(\mathit{exists}(p\text{-}lo, p\text{-}LOR))), \bot, \bot, \bot],\\
\quad \mathit{True})\\[6pt]
\xrightarrow{ISR_{\downarrow(nm)}} \{\text{determine actionality and intentionality}\}\\[4pt]
(\mathit{check}(\mathit{num\text{-}vers}(lo_1) \leq 10).\sim,\\
\quad [\downarrow(\mathit{exists}(p\text{-}lo, p\text{-}LOR))), \texttt{return eval(p-lo in LOR)}, \mathit{exists}(p\text{-}lo, p\text{-}LOR), (p\text{-}lo, p\text{-}LOR))],\\
\quad \mathit{True})\\[6pt]
\xrightarrow{ASR_{out}} \{\text{complete the access action}\}\\[4pt]
(\mathit{check}(\mathit{num\text{-}vers}(lo_1) \leq 10).\sim,\\
\quad [\bot, \bot, \bot, \bot],\\
\quad \mathit{True}.\mathit{True})\\[6pt]
\xrightarrow{ASR_{inp\,1}} \{\text{ideate next input}\}\\[4pt]
(\sim,\\
\quad [\downarrow(\mathit{num\text{-}vers}(p\text{-}lo) \leq 10), \bot, \bot, \bot],\\
\quad \mathit{True})\\[6pt]
\xrightarrow{ISR_{\downarrow(nm)}} \{\text{determine actionality and intentionality}\}\\[4pt]
(\sim,\\
\quad [\downarrow(\mathit{num\text{-}vers}(p\text{-}lo) \leq 10), \begin{array}{l}\texttt{m<-num-vers(p-lo,LOR),}\\ \texttt{return eval(m >= n)}\end{array}, \mathit{num\text{-}vers}(p\text{-}lo, p\text{-}LOR) \geq 10; (p\text{-}lo, p\text{-}LOR))],\\
\quad \mathit{True})\\[6pt]
\xrightarrow{ASR_{out}} \{\text{complete the access action}\}\\[4pt]
(\sim,\\
\quad [\bot, \bot, \bot, \bot],\\
\quad \mathit{True}.\mathit{True}.\mathit{False})\ \{\text{which is the configuration at the end of the initial part the process}\}
\end{array}
$$

Fig. 6. The initial part of the temporal evolution of the constative intentional process of the agent ag.

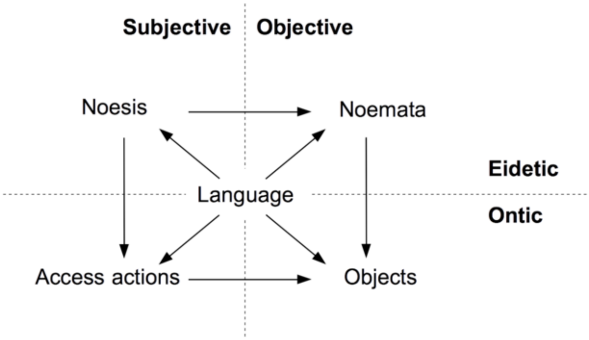

Fig. 7. The place of language in the Husserlian square.

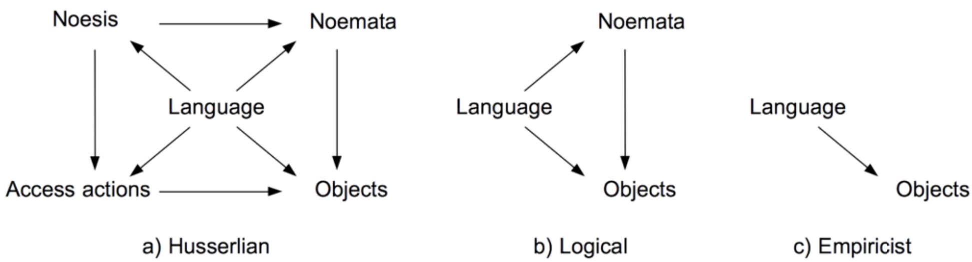

Fig. 8. The alternative semantical perspectives.

Logical semantics, even though reducing the semantical range, compared with the Husserlian semantics, still contemplates the eidetic elements that pertain to language, as one can see in the classical understanding of logical expressions in terms of *connotations* and *denotations*. The same can also be seen in the work on *intensional semantics*, dealing with *intensions* and *extensions* (see, e.g. [7]). Empiricist semantics, on the other hand, strives to drastically restrict language to its sole relation to objects, an option strongly criticized by Husserl [5].

The value of the Husserlian semantical perspective, for works aiming to account for languages in formal models of intentional systems, should be clear, now: both the *eidetic* and the *ontic* semantical aspects of languages receive proper space, as well as their *objective* and *subjective* aspects, as clearly shown in Fig. 7.

5.2 Phenomenological Approaches and the of Intentional Agents

As shown in Fig. 1, the Husserlian phenomenological perspective contemplates four essential aspects of intentionality, namely, the *eidetic* and the *ontic*, as well as the *subjective* and the *objective*.

Thus, not only a complete approach to the intentional systems and the semantics of their languages can be developed, by adopting the Husserlian perspective: a complete approach can also be developed to the structure and operation of the *agents* that concretely realize those intentional systems.

In particular, the fact that subjectiveness is taken into account in the Husserlian perspective, allows a large space for the issue of the *individuality* of those agents.

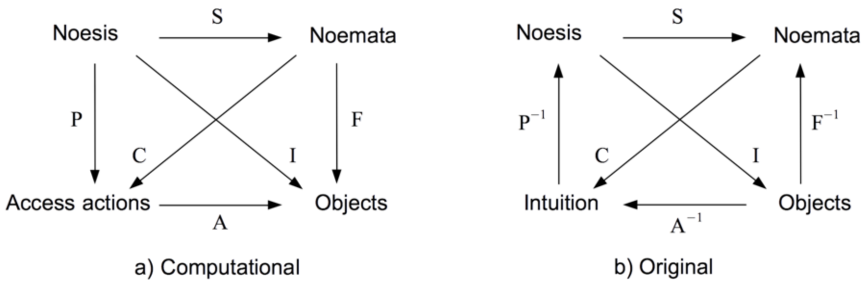

Fig. 9. The computational and the original Husserlian intentional perspectives.

5.3 Phenomenological Reduction and Phenomenological Expansion

Finally, a more general consideration should be made here. There is a contrast between the *computationally oriented approach* to Husserl's intentional perspective, introduced here, and Husserl's original intentional perspective [6]. This is easily seen in Fig. 9.

The whole difference resides in the directions of the relationships of *performance*, *fulfillment* and *access*: P, F and A, in the computational approach, and P^{-1}, F^{-1} and A^{-1}, in the original one.

This is so because, in the original Husserlian approach, the *ontic level* is taken as given, and the phenomenological effort is to achieve the *eidetic level* that corresponds to that ontic level. To allow for that achievement, Husserl elaborated his well-known operation of *phenomenological reduction*, which, given mental access actions and objects, returns their respective noeses and noemata.

In the computational approach proposed here, on the other hand, the aim goes in the opposite direction. We start from the *eidetic level*, and we search for an *ontic level* that can adequately realize to it. The operation thus required is not that of phenomenological reduction, but the inverse one, of *phenomenological expansion*, which Husserl only briefly mentions [6].

In fact, as the problem we have is that of choosing particular access actions and objects to realize the eidetic level, the operation we need here, and that we formalized in the paper, is a choice-based particularization of the phenomenological expansion, an operation that we call *phenomenological realization*.

Thus, the critical issue in determining the adequacy of an ontic structure to an eidetic one, in the computational approach, is the determination of the compatibility C between them, which can not be deduced from P and S (see those functions in the computational square, in Fig. 9).

Husserl, on his turn, had no explicit concern with access actions, generally directed from the agent to the intented objects. On the contrary, what he was concerned solely with was the operation that goes in the opposite direction, namely, *intuition*, with *perception* being the prototypical one [6]. Also, Husserl seemed to have assumed that the eidetic elements (noeses, noemata) obtained by ideation, on the basis of intuition, were always *correct by construction*, as it were. Naturally, then, he seemed to have never been concerned with *adequation* issues, e.g., with the compatibility relationship C.

However, given that Husserl's aimed at a conceptual level (the eidetic) from a distinct one (the ontic), an adequation issue had to appear. Since his *performance* and *fulfillment* functions were taken in the inverse direction (P^{-1} and F^{-1}, in Fig. 9), the critical problem of adequacy appeared not in the determination of the *compatibility relationship* C, but in the determination of the *intentionality relationship*. For, in the formal setting that he established, $\mathrm{I} = \mathrm{F}^{-1} \circ \mathrm{S}$ can not be deductively established, and obtained "by construction".

In fact, Husserl's original *intentionality problem* may be formally construed, in his terms, as follows (see Fig. 9):

> Original Husserlian problem: *Given a theoretical understanding of how noeses relate to noemata (through the arrow* S*), an object and its intuition (given through the access arrow* A^{-1} *between them), to find a noesis (an ideation of an intuition) and corresponding* P^{-1}*, and a noemata (an ideation of the intuited object) and corresponding* F^{-1}*, such that the triangle* $\mathrm{S} = \mathrm{F}^{-1} \circ \mathrm{I}$ *commutes.*

In the computational approach, on the other hand, the intentionality of a noesis is trivially implied by the sense and the fulfillment functions of its noema: one has immediately that $\mathrm{I} = \mathrm{F} \circ \mathrm{S}$.

The general problem that the computational modeling introduced here, then, is another one, namely:

> Computational Husserlian problem: *Given a noesis, a noemata and the theoretical understanding of how they relate (through the arrow* S*), find access actions, accessed objects and arrow* A *such that both the triangles* $\mathrm{I} = \mathrm{F} \circ \mathrm{S}$ *and* $\mathrm{P} = \mathrm{S} \circ \mathrm{C}$ *commute.*

The essential reason for the differences in the approaches can be clearly stated, thus:

- Husserl's aim seems to have been that of grasping the phenomenological aspects of the *natural world*, which is obviously already given to all of us, daily.
- Our aim is to make use of Husserl's results to contribute to computer science and the computational technology, whose main objects are *artifacts* that, after being *specified* and *designed* in an ideal way, have yet to be *constructed* and put into operation in the given natural world.

To use Searle's terminology [9], while *phenomenological reduction* refers to the *mind-world* direction of intentionality, *phenomenological expansion* refers to its *world-mind* direction of intentionality, and it is with the latter that we have been centrally concerned here.

5.4 The Role of Domain Theory

We have not given a treatment for the notion of a object *fulfilling* a noema. Given the type-theoretic nature of the formalization introduced in the paper, it seems only natural to treat treat noemata as ideal specifications of objects, and the notion of *fulfillment* of that ideal specification in terms of a partial *approximation* or *completion* of objects toward their full-fledged forms, as given by their noemata.

Such relation of approximation or completion between *partial* and *total* objects has been extensively studied in *Domain Theory* [4]. It can be assumed, then, that a promising approach to the formal account of the relation between objects and noemata can be attempted on the basis of that theory.

Analogously, the approximation relation between *noeses* and *access actions* can perhaps also be formally accounted for in terms of Domain Theory.

Acknowledgment

The author thanks the anonymous reviewer of the final version for the careful reading of the paper, and for the useful suggestions.

References

[1] Austin, J. L., "How to do Things with Words," Claredon Press, Oxford, 1962.

[2] Enderton, H., "Introduction to Mathematical Logic," Hartcourt, San Diego, 2001.

[3] Fernández, M., "Programming Languages and Operational Semantics: A Concise Overview," Springer, 2014.

[4] Gierz, G., K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove and D. S. Scott, "Continuous Lattices and Domains," Cambridge University Press, Cambridge, 2003.

[5] Husserl, E., "Logical Investigations," Routledge, London, 2001.

[6] Husserl, E., "Ideas I - Ideas for a Pure Phenomenology and Phenomenlogical Philosophy," Hackett, Indianapolis, 2014.

[7] Montague, R., "Formal Philosophy - Selected Papers of Richard Montague," Yale University Press, 1974.

[8] Plotkin, G., *A structural approach to operational semantics*, Technical report, University of Aarhus (1981).

[9] Searle, J. R., "The Construction of Social Reality," The Free Press, New York, 1995.

[10] Smith, D. W., *Phenomenology*, Stanford Encyclopedia of Philosophy (2013), Acessável em: http://plato.stanford.edu/entries/phenomenology/.

[11] Wooldridge, M., "Introduction to Multiagent Systems," Wiley, London, 2011, (2nd. ed.).