

INFORMATION ETHICS GROUP

Oxford University and University of Bari

Levellism and the Method of Abstraction

by

Luciano Floridi and J. W. Sanders

luciano.floridi@philosophy.ox.ac.uk

jeff@comlab.ox.ac.uk



IEG – RESEARCH REPORT 22.11.04

<http://web.comlab.ox.ac.uk/oucl/research/areas/ieg>

Abstract

The use of “levels of abstraction” in philosophical analysis (*levellism*) has recently come under attack. In this paper, we argue that a refined version of *epistemological levellism* should be retained as a fundamental method, which we call *the method of abstraction*. After a brief introduction, in section two we make clear the nature and applicability of the (epistemological) method of levels of abstraction. In section three, we show the fruitfulness of the new method by applying it to five case studies: the concept of agenthood, the Turing test, the definition of emergence, quantum observation and decidable observation. In section four, we further characterise and support the method by distinguishing it from three other forms of “levellism”: (i) levels of organisation; (ii) levels of explanation and (iii) conceptual schemes. In this context, we also briefly address the problems of relativism and antirealism. In the conclusion, we indicate some of the work that lies ahead, two potential limitations of the method and some results that have already been obtained by applying the method to some long-standing philosophical problems.

Keywords

Abstraction, Level of Abstraction, Gradient of Abstraction, Levellism, Observable.

“If we are asking about wine, and looking for the kind of knowledge which is superior to common knowledge, it will hardly be enough for you to say “wine is a liquid thing, which is compressed from grapes, white or red, sweet, intoxicating” and so on. You will have to attempt to investigate and somehow explain its internal substance, showing how it can be seen to be manufactured from spirits, tartar, the distillate, and other ingredients mixed together in such and such quantities and proportions.”

Gassendi, *Fifth Set of Objections to Descartes' Meditations*

1. Introduction

Reality can be studied at different levels, so forms of “levellism” have often been advocated in the past.¹ In the seventies, levellism nicely dovetailed with the computational turn and became a standard approach both in science and in philosophy (Dennett [1971], Mesarovic et al. [1970], Simon [1969], see now Simon [1996], and Wimsatt [1976]). The trend reached its acme at the beginning of the eighties (Marr [1982], Newell [1982]) and since then levellism has enjoyed great popularity² and textbook status (Foster [1992]). However, after decades of useful service, levellism seems to have come under increasing criticism.

Consider the following varieties of levellism available in the literature:

- 1) epistemological, e.g., levels of observation or interpretation of a system (see section 4);
- 2) ontological, e.g., levels (or rather layers) of organization, complexity, or causal interaction etc. of a system;³
- 3) methodological, e.g., levels of interdependence or reducibility among theories about a system; and
- 4) an amalgamation of (1)-(3), e.g., as in Oppenheim and Putnam [1958].

¹ See for example Brown [1916]. Of course the theory of ontological levels and the “chain of being” goes as far back as Plotin and it is the basis of at least one version of the ontological argument.

² The list includes Arbib [1989], Bechtel and Richardson [1993], Egyed and Medvidovic [2000], Gell-Mann [1994], Kelso [1995], Pylyshyn [1984], Salthe [1985].

³ Poli [2001] provides a reconstruction of ontological levellism; more recently, Craver [2004] has analysed ontological levellism, especially in biology and cognitive science, see also Craver [forthcoming].

The current debate on multirealizability in the philosophy of AI and cognitive science has made (3) controversial (Block [1997]). And two recent articles by Heil [2003] and Schaffer [2003] have seriously and convincingly questioned the plausibility of (2). Since criticisms of (2) and (3) end up undermining (4), rumours are that levellism should probably be decommissioned.

In general, we agree with Heil and Schaffer that *ontological levellism* may be untenable, but we also contend that *epistemological levellism* should be retained as a proper method of conceptual analysis, if in a suitably refined version. Fleshing out and defending epistemological levellism is the main task of this paper. We shall proceed in two stages. First, we shall clarify the nature and applicability of what we shall call *the method of (levels of) abstraction*. We shall then distinguish it from other level-based approaches, which may not, and indeed need not, be rescued. Here is a more detailed outline of the paper.

In section two, we provide a definition of the basic concepts fundamental to the method. Although the definitions require some rigour, only the rudiments of mathematical notation are presupposed and all the main concepts are introduced without assuming any previous knowledge. The definitions are illustrated by several intuitive examples, which are designed to familiarise the reader with the method.

In section three, we show how the method of abstraction may be fruitfully applied to several philosophical topics.

In section four, we further characterise and support the method of abstraction by distinguishing it from three forms of “levellism”: (i) ontological levels of organisation; (ii) methodological levels of explanation and (iii) conceptual schemes. In this context we also briefly address the problems of relativism and antirealism.

In the conclusion, we indicate some of the work that lies ahead, two potential limitations of the method and some results that have already been obtained by applying the method to some long-standing philosophical problems in different areas.

Before starting, an acknowledgement of our intellectual debts is in order. Levellism has been of essential importance in science since antiquity. Only more recently has the concept of simulation been used in computer science to relate levels of abstraction (for example de Roeper and Engelhardt [1998], Hoare and He [1998]), to satisfy the

requirement that systems constructed in levels (in order to tame their complexity) function correctly. Our definition of *Gradient of Abstraction* (GoA, see section 2.6) has been inspired by this approach. Indeed, we take as a definition the property established by simulations, namely the conformity of behaviour between levels of abstraction (more on this later). Necessarily, our definition of GoA remains mathematical and for this reason we do not follow through all details in the examples. However, we hope that the discussion of Turing's imitation game in section 3.2 will make clear not just the use of levels of abstraction but also their conformance in a GoA.

2. Definitions and preliminary examples

In this section, we define six concepts (“typed variable”, “observable”, “level of abstraction”, “behaviour”, “moderated level of abstraction” and “gradient of abstraction”) and then the “method of abstraction” based on them.

2.1. Typed variable

A variable is a symbol that acts as a place-holder for an unknown or changeable referent.

A “typed variable” is a variable qualified to hold only a declared kind of data.

Definition. A *typed variable* is a uniquely-named conceptual entity (the *variable*) and a set, called its *type*, consisting of all the values that the entity may take. Two typed variables are regarded as *equal* if and only if their variables have the same name and their types are equal as sets. A variable that cannot be assigned well-defined values is said to constitute an *ill-typed variable* (see the example in section 2.3).

When required, we shall write $x:X$ to mean that x is a variable of type X . Positing a typed variable means taking an important decision about how its component variable is to be conceived. We shall be in a position to appreciate this point better after the next definition.

2.2. Observable

The notion of an “observable” is common in science, occurring whenever a (theoretical) model is constructed. Although the way in which the features of the model correspond to the system being modelled is usually left implicit in the process of modelling, it is important here to make that correspondence explicit. We shall use the word “system” to stand for the object of study. This may indeed be what would normally be described as a system in science or engineering, but it may also be a domain of discourse, of analysis or of conceptual speculation: a purely semantic system, as it were.

Definition. An *observable* is an interpreted typed variable, that is, a typed variable together with a statement of what feature of the system under consideration it represents. Two observables are regarded as *equal* if and only if their typed variables are equal, they model the same feature and, in that context, one takes a given value if and only if the other does.

Being an abstraction, an observable is not necessarily meant to result from quantitative measurement or even empirical perception. The “feature of the system under consideration” might be a physical magnitude – we shall return to this point in section 3.4, when talking about quantum observation – but it might also be an artefact of a conceptual model, constructed entirely for the purpose of analysis.

An observable, being a typed variable, has specifically determined possible values. In particular:

Definition. An observable is called *discrete* if and only if its type has only finitely many possible values; otherwise it is called *analogue*.⁴

In this paper, we are interested in observables as a means of describing behaviour at a precisely qualified (though seldom numerical) level of abstraction; in general, several observables will be employed.

⁴ The distinction is really a matter of topology rather than cardinality. However, this definition serves our present purposes.

2.3. Five examples

Let us now consider some simple examples.

1) Suppose we wish to study some physical human attributes. To do so we, in Oxford, introduce a variable, h , whose type consists of rational numbers. The typed variable h becomes an (analogue) observable once we decide that the variable h represents the height of a person, using the Imperial system (feet and parts thereof). To explain the definition of equality of observables, suppose that our colleague, in Rome, also interested in observing human physical attributes, defines the same typed variable but declares that it represents height in metres and parts thereof. Our typed variables are the same, but they differ as observables: for a given person, the two variables take different representing values. This example shows the importance of making clear the interpretation by which a typed variable becomes an observable.

2) Consider next an example of an ill-typed variable. Suppose we are interested in the rôles played by people in some community; we could not introduce an observable standing for those beauticians who depilate just those people who do not depilate themselves, for it is well-known that such a variable would not be well typed (Russell [1902]). Similarly, each of the standard antinomies (Hughes and Brecht [1976]) reflects an ill-typed variable. Of course, the modeller is at liberty to choose whatever type befits the application, and if that involves a potential antinomy then the appropriate type might turn out to be a non-well-founded set (Barwise and Etchemendy [1987]). However, in this paper we shall operate entirely within the boundaries of standard naive set theory.

3) Suppose we follow Gassendi and wish to analyse wine. Observables relating to tasting wine include the attributes that commonly appear on “tasting sheets”: *nose* (representing bouquet), *legs* or *tears* (viscosity), *robe* (peripheral colour), *colour*, *clarity*, *sweetness*, *acidity*, *fruit*, *tannicity*, *length* and so on, each with a determined type.⁵ If two wine tasters choose different types for, say, *colour* (as is usually the case) then the observables

⁵ Despite a recent trend towards numeric values, these have not been standardised and so we leave to the reader the pleasant task of contemplating appropriate types; for a secondary source of inspiration we refer to tasting-related entries in Robinson [1994].

are different, despite the fact that their variables have the same name and represent the same feature in reality. Indeed, as they have different types they are not even equal as typed variables.

Information about how wine quality is perceived to vary with time – how the wine “ages” (Robinson [1989]) – is important for the running of a cellar. An appropriate observable is the typed variable a , which is a function associating to each year y : *Years* a perceived quality $a(y)$: *Quality*, where the types *Years* and *Quality* may be assumed to have been previously defined. Thus, a is a function from *Years* to *Quality*, written $a: \textit{Time} \rightarrow \textit{Quality}$. This example shows that, in general, types are constructed from more basic types, and that observables may correspond to operations, taking input and yielding output. Indeed, an observable may be of arbitrarily complex type.

4) The definition of an observable reflects a particular view or attitude towards the entity being studied. Most commonly, it corresponds to a simplification, in which case nondeterminism, not exhibited by the entity itself, may arise. The method is successful when the entity can be understood by combining the simplifications. Let us consider another example.

In observing a game of chess, we would expect to record the moves of the game.⁶ Other observables might include the time taken per move; the body language of the players; and so on. Suppose we are able to view the chessboard by looking just along *files* (the columns stretching from player to player). When we play “files-chess”, we are unable to see the ranks (the parallel rows between the players) or the individual squares. Files cannot sensibly be attributed a colour black or white, but each may be observed to be occupied by a set of pieces (namely those that appear along that file), identified in the usual way (king, queen and so forth). In “files-chess”, a move may be observed by the effect it has on the file of the piece being moved. For example, a knight moves one or two files either left or right from its starting file; a bishop is indistinguishable from a rook, which moves along a rank; and a rook that moves along a file appears to remain

⁶ This is done by recording the history of the game: move by move the state of each piece on the board is recorded – in English algebraic notation – by rank and file, as are recorded the piece being moved and the consequences of the move.

stationary. Whether or not a move results in a piece being captured, appears to be nondeterministic. “Files-chess” seems to be an almost random game.

Whilst the “underlying” game is virtually impossible to reconstruct, each state of the game and each move (i.e., each operation on the state of the game) can be “tracked” with this dimensionally-impooverished family of observables. If one then takes a second view, corresponding instead to rank, we obtain “ranks-chess”. Once the two views are combined, the original game of chess can be recovered, since each state is determined by its rank and file projections, and similarly for each move. The two disjoint observations together (“files-chess” + “ranks-chess”) reveal the underlying game.

5) The degree to which a type is appropriate depends on its context and use. For example, to describe the state of a traffic light in Rome we might decide to consider an observable *colour* of type $\{red, amber, green\}$ that corresponds to the colour indicated by the light. This option abstracts the length of time for which the particular colour has been displayed, the brightness of the light, the height of the traffic light, and so on. This is why the choice of type corresponds to a decision about how the phenomenon is to be regarded. To specify such a traffic light for the purpose of construction, a more appropriate type would comprise a numerical measure of wavelength (see section 2.6). Furthermore, if we are in Oxford, the type of colour would be a little more complex, since – in addition to red, amber and green – red and amber are displayed simultaneously for part of the cycle. So, an appropriate type would be $\{red, amber, green, red-amber\}$.

2.4. Level of abstraction

We are now ready to introduce the basic concept of *level of abstraction* (LoA).

Any collection of typed variables can, in principle, be combined into a single “vector” observable, whose type is the Cartesian product of the types of the constituent variables. In the wine example, the type *Quality* might be chosen to consist of the Cartesian product of the types *Nose*, *Robe*, *Colour*, *Acidity*, *Fruit* and *Length*. The result would be a single, more complex, observable. In practice, however, such vectorisation is unwieldy, since the expression of a constraint on just some of the observables would require projection

notation to single out those observables from the vector. Instead, we shall base our approach on a *collection* of observables, that is, a level of abstraction:

Definition. A *level of abstraction (LoA)* is a finite but non-empty set of observables. No order is assigned to the observables, which are expected to be the building blocks in a theory characterised by their very definition. A LoA is called *discrete* (respectively *analogue*) if and only if all its observables are discrete (respectively analogue); otherwise it is called *hybrid*.

Consider the wine example. Different LoAs may be appropriate for different purposes. To evaluate a wine, the “tasting LoA”, consisting of observables like those mentioned in the previous section, would be relevant. For the purpose of ordering wine, a “purchasing LoA” – containing observables like *maker, region, vintage, supplier, quantity, price*, and so on – would be appropriate; but here the “tasting LoA” would be irrelevant. For the purpose of storing and serving wine – the “cellaring LoA” - containing observables for *maker, type of wine, drinking window, serving temperature, decanting time, alcohol level, food matchings, quantity remaining in the cellar*, and so on – would be relevant.

The traditional sciences tend to be dominated by analogue LoAs, the humanities and information science by discrete LoAs and mathematics by hybrid LoAs. We are about to see why the resulting theories are fundamentally different.

2.5. Behaviour

The definition of observables is only the first step in studying a system at a given LoA. The second step consists in deciding what relationships hold between the observables. This, in turn, requires the introduction of the concept of system “behaviour”. We shall see that it is the fundamentally different ways of describing behaviour in analogue and discrete systems that account for the differences in the resulting theories.

Not all values exhibited by combinations of observables in a LoA may be realised by the system being modelled. For example, if the four traffic lights at an intersection are modelled by four observables, each representing the colour of a light, the lights cannot in fact all be green together (assuming they work properly). In other words, the combination in which each observable is green cannot be realised in the system being modelled, although the types chosen allow it. Similarly, the choice of types corresponding to a rank-

and-file description of a game of chess allows any piece to be placed on any square, but in the actual game two pieces cannot occupy the same square simultaneously.

Some technique is therefore required to describe those combinations of observable values that are actually acceptable. The most general method is simply to describe all the allowed combinations of values. Such a description is determined by a predicate, whose allowed combinations of values we call the “system behaviours”.

Definition. A *behaviour* of a system, at a given LoA, is defined to consist of a predicate whose free variables are observables at that LoA. The substitutions of values for observables that make the predicate true are called the *system behaviours*. A *moderated LoA* is defined to consist of a LoA together with a behaviour at that LoA.

Consider two previous examples. In reality, human height does not take arbitrary rational values, for it is always positive and bounded above by (say) nine feet. The variable h , representing height, is therefore constrained to reflect reality by defining its behaviour to consist of the predicate $0 < h < 9$, in which case any value of h in that interval is a “system” behaviour. Likewise, wine too is not realistically described by arbitrary combinations of the aforementioned observables. For instance, it cannot be both white and highly tannic.

Since Newton and Leibniz, the behaviours of the analogue observables, studied in science, have typically been described by differential equations. A small change in one observable results in a small, quantified change in the overall system behaviour. Accordingly, it is the rates at which those smooth observables vary which is most conveniently described.⁷ The desired behaviour of the system then consists of the solution of the differential equations. However, this is a special case of a predicate: the predicate holds at just those values satisfying the differential equation. If a complex system is approximated by simpler systems, then the differential calculus provides a supporting method for quantifying the approximation.

⁷ It is interesting to note that the catastrophes of *chaos theory* are not smooth; although they do appear so when extra observables are added, taking the behaviour into a smooth curve on a higher-dimensional manifold. Typically, chaotic models are weaker than traditional models, their observables merely reflecting *average* or *long-term* behaviour. The nature of the models is clarified by making explicit the LoA.

The use of predicates to demarcate system behaviour is essential in any (nontrivial) analysis of discrete systems because in the latter no such continuity holds: the change of an observable by a single value may result in a radical and arbitrary change in system behaviour. Yet, complexity demands some kind of comprehension of the system in terms of simple approximations. When this is possible, the approximating behaviours are described exactly, by a predicate, at a given LoA, and it is the LoAs that vary, becoming more comprehensive and embracing more detailed behaviours, until the final LoA accounts for the desired behaviours. Thus, the formalism provided by the method of abstraction can be seen as doing for discrete systems what differential calculus has traditionally done for analogue systems.

Likewise, the use of predicates is essential in subjects like information and computer science, where discrete observables are paramount and hence predicates are required to describe a system behaviour. In particular, state-based methods like *Z* (Hayes and Flinn [1993], Spivey [1992]) provide notation for structuring complex observables and behaviours in terms of simpler ones. Their primary concern is with the syntax for expressing those predicates, an issue we shall try to avoid in this paper by stating predicates informally.

The time has come now to combine approximating, moderated LoAs to form the primary concept of the method of abstraction.

2.6. Gradient of abstraction

For a given (empirical or conceptual) system or feature, different LoAs correspond to different representations or views. A *Gradient of Abstractions* (GoA) is a formalism defined to facilitate discussion of discrete systems over a range of LoAs. Whilst a LoA formalises the scope or granularity of a single model, a GoA provides a way of varying the LoA in order to make observations at differing levels of abstraction.

For example, in evaluating wine we might be interested in the GoA consisting of the “tasting” and “purchasing” LoAs, whilst in managing a cellar we might be interested in the GoA consisting of the “cellaring” LoA together with a sequence of annual results of observation using the “tasting” LoA.

In general, the observations at each LoA must be explicitly related to those at the others; to do so, we use a family of relations between the LoAs. For this, we need to recall some (standard) preliminary notation.

Notation. A *relation* R from a set A to a set C is a subset of the Cartesian product $A \times C$. R is thought of as relating just those pairs (a, c) that belong to the relation. The *reverse* of R is its mirror image: $\{(c, a) \mid (a, c) \in R\}$. A relation R from A to C translates any predicate p on A to the predicate $P_R(p)$ on C that holds at just those $c:C$, which are the image through R of some $a:A$ satisfying p

$$P_R(p)(c) = \exists a: A \ R(a,c) \wedge p(a).$$

We have finally come to the main definition of the paper.

Definition. A *gradient of abstractions*, GoA , is defined to consist of a finite set⁸ $\{L_i \mid 0 \leq i < n\}$ of moderated LoAs L_i and a family of relations $R_{i,j} \subseteq L_i \times L_j$, for $0 \leq i \neq j < n$, relating the observables of each pair L_i and L_j of distinct LoAs in such a way that:

1. the relationships are inverse: for $i \neq j$, $R_{i,j}$ is the reverse of $R_{j,i}$
2. the behaviour p_j at L_j is at least as strong as the translated behaviour $P_{R_{i,j}}(p_i)$

$$p_j \Rightarrow P_{R_{i,j}}(p_i). \tag{1}$$

Two GoAs are regarded as *equal* if and only if they have the same moderated LoAs (i.e., the same LoAs and moderating behaviours) and their families of relations are equal. A GoA is called *discrete* if and only if all its constituent LoAs are discrete.

Condition (1) means that the behaviour moderating each lower LoA is *consistent* with that specified by a higher LoA. Without it, the behaviours of the various LoAs constituting a GoA would have no connection to each other. A special case, to be elaborated below in the definition of “nestedness”, helps to clarify the point.

If one LoA L_i extends another L_j by adding new observables, then the relation $R_{i,j}$ is the inclusion of the observables of L_i in those of L_j and (1) reduces to this: the

⁸ The case of infinite sets has application to analogue systems but is not considered here.

constraints imposed on the observables at LoA L_i remain true at LoA L_j , where “new” observables lie outside the range of $R_{i,j}$.

A GoA whose sequence contains just one element evidently reduces to a single LoA. So our definition of “LoA” is subsumed by that of “GoA”.

The consistency conditions imposed by the relations $R_{i,j}$ are in general quite weak. It is possible, though of little help in practice, to define GoAs in which the relations connect the LoAs cyclically. Of much more use are the following two important kinds of GoA: “disjoint” GoAs (whose views are complementary) and “nested” GoAs (whose views provide successively more information). Before defining them we need a little further notation.

Notation. We recall that a *function* f from a set C to a set A is a relation, i.e., a subset of the Cartesian product $C \times A$, which is single-valued

$$\forall c:C \quad \forall a, a':A \quad ((c,a) \in f \wedge (c,a') \in f) \Rightarrow a = a'$$

(this means that the notation $f(c) = a$ is a well-defined alternative to $(c,a) \in f$, and total

$$\forall c:C \quad \exists a:A \quad f(c) = a$$

(this means that $f(c)$ is defined for each $c:C$). A function is called *surjective* if and only if every element in the target set lies in the range of the function:

$$\forall a:A \quad \exists c:C \quad f(c) = a.$$

Definition. A GoA is called *disjoint* if and only if the L_i are pairwise disjoint (i.e., taken two at a time, they have no observable in common) and the relations are all empty. It is called *nested* if and only if the only nonempty relations are those between L_i and L_{i+1} , for each $0 \leq i < n-1$, and moreover the reverse of each $R_{i, i+1}$ is a surjective function from the observables of L_{i+1} to those of L_i .

A disjoint GoA is chosen to describe a system as the combination of several non-overlapping components. This is useful when different aspects of the system behaviour

are better modelled as being determined by the values of distinct observables. This case is rather simplistic, since the LoAs are more typically tied together by common observations. For example, the services in a domestic dwelling may be represented by LoAs for electricity, plumbing, telephone, security and gas. Without going into detail about the constituent observables, it is easy to see that, in an accurate representation, the electrical and plumbing LoAs would overlap whilst the telephone and plumbing would not.

A nested GoA (see Fig.1) is chosen to describe a complex system exactly at each level of abstraction and incrementally more accurately. The condition that the functions be surjective means that any abstract observation has at least one concrete counterpart. As a result, the translation functions cannot overlook any behaviour at an abstract LoA: behaviours lying outside the range of a function translate to the predicate *false*. The condition that the reversed relations be functions means that each observation at a concrete LoA comes from at most one observation at a more abstract LoA (although the converse fails in general, allowing one abstract observable to be refined by many concrete observables). As a result the translation functions become simpler.

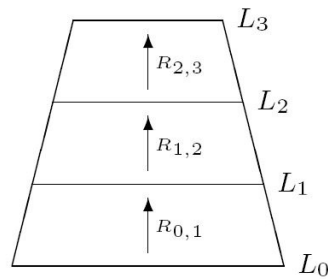


Fig. 1 Nested GoA with four Levels of Abstraction

For example, the case of a traffic light which is observed to have colour *colour* of type $\{red, amber, green\}$ is captured by a LoA, L_0 , having that single observable. If we wish to be more precise about colour, perhaps for the purpose of constructing a new traffic light, we might consider a second LoA, L_1 , having the variable wl whose type is a positive real number corresponding to the wavelength of the colour. To determine the behaviour of L_1 , Suppose that constants $\lambda_{red} < \lambda_{red}'$ delimit the wavelength of red, and

similarly for amber and green. Then the behaviour of L_1 is simply this predicate with free variable wl

$$(\lambda_{red} \leq wl \leq \lambda_{red}') \vee (\lambda_{amber} \leq wl \leq \lambda_{amber}') \vee (\lambda_{green} \leq wl \leq \lambda_{green}').$$

The sequence consisting of the LoA L_0 and the moderated LoA L_1 forms a nested GoA. Informally, the smaller, abstract, type $\{red, amber, green\}$ is a projection of the larger. The relevant relation associates to each value $c:\{red, amber, green\}$ a band of wavelengths perceived as that colour. Formally, $R(colour, wl)$ is defined to hold if and only if, for each $c:\{red, amber, green\}$,

$$colour = c \leftrightarrow \lambda_c \leq wl \leq \lambda_c'.$$

In the wine example, a first LoA might be defined to consist of the variable “kind” having type consisting of *red, white, rose* under the obvious representation. A second LoA might be defined to consist of the observable “kind” having type

$$\{stillred, sparklingred, stillwhite, sparklingwhite, stillrose, sparklingrose\}.$$

Although the second type does not contain the first, it produces greater resolution under the obvious projection relation. Thus, the GoA consisting of those two LoAs is nested.

Those two important forms of GoA – disjoint and nested – are in fact theoretically interchangeable. For if A and B are disjoint sets then A and their union $A \cup B$ are increasing sets and the former is embedded in the latter. Thus, a disjoint GoA can be converted to a nested one. Conversely, if A and B are increasing sets with the former embedded in the latter, then A and the set difference $A \setminus B$ are disjoint sets. Thus, a nested GoA can be converted to a disjoint one.

Following the technique used to define a nested GoA, it is possible to define less restricted but still hierarchical GoAs. Important examples include tree-like structures, of which our nested GoAs are a special, linear case.

For theoretical purposes, the information captured in a GoA can be expressed equivalently as a single LoA of more complicated type, namely one whose single LoA has type equal to the sequence of the LoAs of the complex interface. However, the current definition is better suited to application.

2.7. The method of abstraction

Models are the outcome of the analysis of a system, developed at some LoA(s) for some purpose. An important contribution of these ideas is to make precise the commitment to a LoA/GoA before further elaborating a theory. We call this the *method of abstraction*. Three main advantages of the method can be highlighted here.

First, specifying the LoA means clarifying, from the outset, the range of questions that (a) can be meaningfully asked and (b) are answerable in principle. One might think of the input of a LoA as consisting of the system under analysis, comprising a set of *data*; its output is a *model* of the system, comprising *information*. The quantity of information in a model varies with the LoA: a lower LoA, of greater resolution or finer granularity, produces a model that contains more information than a model produced at a higher, or more abstract, LoA. Thus, a given LoA provides a quantified commitment to the kind and amount of information that can be “extracted” from a system. The choice of a LoA pre-determines the type and quantity of data that can be considered and hence the information that can be contained in the model. So, knowing at which LoA the system is being analysed is indispensable, for it means knowing the scope and limits of the model being developed.

Second, being explicit about the LoA adopted provides a healthy antidote to ambiguities, equivocations and other fallacies or errors due to level-shifting, such as Aristotle’s “metabasis eis allo genos” (shifting from one genus to another) and Ryle’s “category-mistakes”.

Third, by stating its LoA, a theory is forced to make explicit and clarify its ontological commitment. The ontological commitment of a theory is best understood by distinguishing between a *committing* and a *committed* component. A theory commits itself ontologically by opting for a specific LoA. Compare this to the case in which one has chosen a specific kind (better not speak of “model” here, to avoid confusion) of car

but has not bought one yet. A theory becomes ontologically committed in full through its model, which is therefore the bearer of the specific commitment. The analogy here is with the specific car one has actually bought. So LoAs commit a theory to types, while their ensuing models commit it to the corresponding tokens.

3. Case studies

The advantages of the method of abstraction can be shown by applying it to some more substantial examples. In this section, we shall consider five of them: agenthood, the Turing test, emergence, quantum observation and decidable observation.

3.1. Agents

An *agent A* can be thought of (see Floridi and Sanders [2004]) as a *transition system* (i.e., a system of states and transitions between them) that is *interactive* (i.e., *A* can respond to a stimulus by a change of state), *autonomous* (i.e., *A* is able to change state without any stimulus) and *adaptable* (i.e., *A* is able to change the transition rules by which it changes state). However, each of these properties, and hence the definition of agenthood, makes sense only at a prescribed LoA. Consider the following examples.

1) Whether or not a rock is deemed to be interactive depends on the length of time and level of detail of observation. Over a long period it erodes and hence changes state. By day, it absorbs solar radiation, which it emits at night. However, if one relies on observables resulting from scrutiny over a period of ten seconds by the naked eye from ten metres, a stone can be deemed not to be interactive.

2) If the LoA adopted abstracts gravity and resistance, a swinging pendulum appears to be autonomous but neither interactive nor adaptive. By extending the LoA to incorporate air resistance, it becomes adaptive. By observing also the whistling sound it makes with the air, it becomes interactive.

3) If a piece of software that exhibits machine learning (Mitchell [1997]) is studied at a LoA that registers its interactions with its environment, then the software will appear interactive, autonomous and adaptive, i.e., to be an agent. However, if the program code is revealed, then the software is shown to be simply following rules and hence not to be adaptive. Those two LoAs are at variance. One reflects the “open source” view of

software: the user has access to the code. The other reflects the commercial view: although the user has bought the software and can use it at will, she has no access to the code. At stake is whether or not the software forms an (artificial) agent. For further examples we refer to Floridi and Sanders [2004].

3.2. The Turing test

Turing [1950] took the incisive step of arguing that the ability to think (“intelligence”) can be satisfactorily characterised by means of a test, rather than by explicit definition. In retrospect, that step may seem a small one. After all, we are quite familiar with contexts in which no explicit definition is possible or sensible. Society makes no attempt to characterise what it means to be an acceptable driver in terms of vision, response times, coordination, experience and other physical attributes. Instead, it relies on a driving test. Likewise, society does not attempt to define what it means for a school student to have reached an acceptable academic standard by the end of school; it relies on final school examinations. Nevertheless, incisive that step certainly must have been in view of the vast number of attempts (even to this day) to characterise intelligence explicitly.⁹

Opponents of Turing’s approach usually object that his test functions at the wrong level of abstraction: perhaps it ought to include a component of creativity, of spontaneity, of embodiment, of emotional involvement and so forth. However, without concepts like those introduced above, it is hard to make one’s objections precise or defend Turing’s approach. So let us see, first, how the Turing test can be expressed using phenomenological LoAs and, second, how it can be analysed using conceptual LoAs.

3.2.1. Turing’s imitation game

Let us start, as did Turing, by considering an imitation game, in which a man *A* and a woman *B* are placed in a room separate from an interrogator *C*, who communicates with each by teleprinter (or these days by computer). *C* poses questions to *A* and *B*, who are known only as *X* and *Y*. *C*’s task is to identify $X = A$ and $Y = B$ or, conversely, $X = B$ and $Y = A$, by considering their responses.

⁹ On the history of the Turing Test, see Shieber [2004].

We might describe this scenario by taking a first, extremely abstract, LoA to reflect just the correctness of C 's identification. The LoA L_0 consists of a single variable ans of type $\{right, wrong\}$, which becomes an observable under the correspondence: ans takes the value $right$ if C is correct and the value $wrong$ if C is incorrect. By choosing this LoA, we intentionally abstract the actual answer (whether X was A or B), the questions and answers, response times, and so on, in order to capture just the outcome of the imitation game.

We might reveal C 's actual identification by defining a second, disjoint, LoA L_1 whose single variable, Z , is of type $\{(A, B), (B, A)\}$, which is made into an observable under the correspondence that the first component of Z is the putative identity of X and the second component that of Y . Combining the two LoAs L_0 and L_1 gives a disjoint GoA.

Of course, there are alternative approaches, which is why it is important to be precise about the one adopted. One might define a GoA by replacing the LoA L_1 with a LoA containing two observables, the first corresponding to the identity of X and the second corresponding to the identity of Y . This would be more involved, since each would have type $\{A, B\}$ and one would have to moderate it with the behaviour that the values of X and Y differ. Our choice of L_1 avoids this complication by building that behaviour into the type of Z but, with several observables, in general such moderating behaviours cannot be circumvented.

In order to model C 's questions, the addressees and their responses, we define a third LoA, L_2 . Let Q and R denote the sets of possible (well-posed) questions and responses respectively (an example where the type of text strings may be considered appropriate). Then each "question, addressee and response" triplet is a variable whose type is the Cartesian product of Q , $\{X, Y\}$ and R . It becomes an observable under the correspondence just established. The observable we seek now consists of a sequence (of arbitrary but finite length) of such triplets, corresponding to the sequence of interactions in temporal order; and L_2 contains that single observable (an alternative would be to have an observable for the number of questions, an observable for each question and an observable for each response). L_2 can be added to either GoA T or T' to obtain a GoA which is still disjoint but has higher resolution.

More detailed LoAs are possible and easy to define but, following Turing, we stop here. It is clear that any discussion of the imitation game can be accurately “calibrated”, according to its level of abstraction, with a GoA.

3.2.2. Turing’s test analysed

In the Turing test, A is replaced by a “machine” (nowadays a “computer”). Turing proposed that the question “can machines think?” be replaced by the question: “will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?”.

These days the test is normally stripped of its sex-specific nature and the interrogator is simply asked to determine the human from the machine.¹⁰ Appropriate GoAs are defined as above, but with A representing a computer and B a human.

Although Turing did not make it explicit, the phrase “as often” in his description implies repetition of the test, and hence a conclusion reached by statistical analysis. Suppose that C initiates a pair of question/answer sessions of the type used in the imitation game. A list of questions is put in two situations, one containing a man and a woman, the other containing a machine and a woman. We suppose that the answers in the first situation are of type A_1 and those in the second of type A_2 , thus avoiding here the question of whether or not $A_1 = A_2$. As before, C makes an identification. The appropriate LoA has type, call it J , equal to the Cartesian product of the type $\{(A, B), (B, A)\}$ and the type of all sequences of elements of the type $Q \times \{X, Y\} \times A_1 \times A_2$.

The observable corresponding to repetition of that situation j times, though not necessarily with the same questions, has type consisting of the Cartesian product of j -many copies of type J , namely J^j . The LoA incorporating that observation plus the answer to the ultimate question is then the Cartesian product of the type J^j and the type $\{right, wrong\}$. Likewise, a more complex type can be constructed to reveal the nature of the statistical test; in this case too, we follow Turing and omit the details.

¹⁰ For a summary of the Turing test today, and its incarnation in competitive form (the Loebner prize), see Moor [2001].

3.2.3. Turing's test discussed

The previous two sections have shown how to formalise the Turing test using phenomenologically motivated GoAs, but the method of abstraction can be used to discuss and compare variations on the test. Indeed, it is difficult to imagine how such an analysis could be formulated accurately without a concept equivalent to LoA. Of course, details of the LoAs need not be given as long as it is clear that they could be.

Turing couched his test in terms of a single human interrogator. In the Loebner test (Moor [2001]), the interrogation is carried out by a panel of humans interacting via computer interface in real time. Alternatively, the interrogator could be a machine; or instead of real-time interaction, a list of pre-arranged questions might be left, and the list of answers returned to the interrogator; or instead of serial questions and answers the interrogator might hold a “general conversation” (Moor [2001]). Each alternative modifies the power of the test. All can be formalised by defining different GoAs.

The Turing test might be adapted to target abilities other than “thinking”, like: interpretation of text, game playing, puzzle solving, spatial reasoning, creativity, and so on. Contemplating the possible GoAs provides a way to formalise the variant test clearly and elegantly, and promotes simple comparison with contending treatments.

3.3. Emergence

The method of abstraction is ideally suited to the study of systems so complex that they are best understood stepwise, that is, by their gradual disclosure at increasingly fine levels of abstraction.¹¹

A key concept in such an approach to complex systems is that of “emergent behaviour”, that is, behaviour that arises in the move from one LoA to a finer level.¹²

¹¹ Gell-Mann [1994] has suggested that the study of such phenomena be called *plectics*. He introduces it using an idea he calls *granularity*, which is conveniently formalised by LoA. The concept of emergence is coupled not only to that of complexity but also to that of reduction. However, for reasons of space, we shall limit ourselves here only to the first, leaving the analysis of the other two to the reader. Damper [2000] discusses reductionism and emergent properties from a LoA perspective.

¹² See Hendriks-Jansen [1989], section 2 for a discussion.

3.3.1. Tossing a coin

Emergence, not surprisingly for such an important concept, can take various forms (Cariani [1991], Nagel [1974]). Epistemologically, the concept refers to the fact that “properties at higher levels are not necessarily predictable from properties at lower levels”.¹³ Here, we shall be concerned only with the concept of emergence, not with its empirical counterparts. We shall see that it is neatly captured using a GoA containing two nested LoAs.¹⁴ Let us begin with an example.

The process of tossing a coin may be modelled abstractly, with an observable *outcome* of type $\{head, tail\}$ corresponding to the side of the coin that faces upwards after the toss. This LoA abstracts any other result, like the coin’s landing on its edge or becoming lost, and all other features, like type of agent tossing the coin, manner of tossing, number of spins, time taken and so on. In particular, it models just one toss of the coin and so it cannot account for the “fairness” of the coin, which can be revealed statistically only after a large number of tosses.

Now, suppose we wish to model the repetition of that process with the explicit aim of discussing the fairness of a coin. We introduce a more concrete LoA, whose observables are: a natural number n , corresponding to the number of tosses of the coin; and a list of n values from $\{head, tail\}$, corresponding to successive tosses as modelled above. At this second LoA, it is possible to assess the fairness of the coin – using standard statistics, for example – on the basis of the frequency of the outcomes. So this is an example of an emergence property in the following sense: in many repeated tosses of the coin, the more abstract model applies toss by toss, but does not allow frequency of outcome to be observed, as it is in the finer model. We say that the notion of the coin’s fairness is emergent at the finer LoA. We are now ready for a definition.

3.3.2. Emergence defined

Suppose that a system S is under consideration using a nested GoA consisting of two LoAs. Let us also assume that the more abstract LoA, A , is moderated by behaviour p_A , which describes the abstract view of S , and that the more concrete LoA, C , is moderated

¹³ Hendriks-Jansen [1989], p. 283; note that, in the quotation, “lower levels” are more abstract and “higher levels” more detailed or concrete.

¹⁴ For a similar line of reasoning see Damper [2000].

by behaviour p_C , which describes the concrete view of S . The abstract and concrete observables are related by the total and one-to-many relation $R_{A,C}$. Recall that a behaviour of the system S at LoA C is a tuple of values for the observables of C that satisfies p_C .

Definition. A behaviour of a system S at LoA C is said to be *emergent* (with respect to that nested GoA) if and only if its translation under the relation $R_{A,C}$ fails to satisfy p_A .

Emergence is said to hold in the GoA if and only if there is some emergent behaviour. There is frequently confusion about emergence. This is scarcely surprising since, without the notion of LoA, the various levels at which a system is discussed cannot be formalised. Emergence arises typically because the concrete LoA embodies a “mechanism”, or rule for determining an observable, which has been overlooked at the abstract LoA, usually quite deliberately, in order to gain simplicity at the cost of detail. Frequently, the breakthrough in understanding some complex phenomenon comes by accounting for emergent behaviour, and this results from considering the process by which it occurs, rather than by taking a more static view of the ingredients or components involved.¹⁵ In the coin example, we have avoided incorporating any mechanism, but any of the multitude of pseudo-random number generators could be used to generate lists of *head* and *tail* and hence to account for the emergent phenomenon (Knuth [1997]).

3.3.3. Process observed

The majority of observables considered so far have been “static”, but operations too constitute vital observables. Indeed, the importance of “process” may be indicated by the example of a sponge cake. With only the ingredients as observables (i.e., the amount of each ingredient) the sponge-like nature of the cake is, as many a novice cook has found, emergent. However, if the manner of aeration (a variable indicating the aeration effect of bicarbonate of soda under the right conditions) is also an observable, then sponginess is explicable. In other words, the behaviour of sponginess emerges at the finer level of abstraction only.

¹⁵ This is well documented by Emmeche et al. [1997], who, however, go on to analyse emergence in terms of ontological or *de re* levels.

3.4. Quantum observation

It is remarkable that the disparate disciplines of quantum mechanics and social anthropology share a fundamental feature: in each, observation inevitably involves interference. Observing a quantum or anthropological system is possible only at the expense of a change to the system. By contrast, our definition of observable makes no assumptions about how the system is (capable of being) measured (effectively) in practice. In this section we address this issue.

In quantum mechanics, “observable” has a much more restricted meaning than in this paper. There it is to be distinguished from the state that is posited to exist in order to explain the frequency with which observables take their values. Such “beables” (Bell [1987]) are for us also observables as is, for that matter, the frequency with which an observable takes on its values. The latter might be regarded as unachievable in practice, since any finite number of readings can achieve only an approximation to it, but that need be of no concern to us. Our only requirement is that an observable be well-typed. When desired, the stricter “observation as measurement” from quantum mechanics can be modelled as a certain kind of observation in our sense: the change in behaviour associated with an “observation as measurement” event is simply specified to conform to the uncertainty principle. The same holds for the constraint of quantum mechanics that only certain (i.e., commuting) observables may be measured simultaneously: whilst two events, say A and B, may be observed independently, their simultaneous observation constitutes a third event, AB say, with the different behavioural consequences dictated by quantum mechanics.

3.5. Decidable observation

In the theory of computation, an observable is called decidable, or effective, if and only if its behaviour is given by a computable function. For example, it is known to be undecidable whether or not a program terminates i.e., there is no algorithm for its determination (Boolos et al. [2002]). We make no assumption about the decidability of an observable. The reason is simple. The field of *Formal Methods* within computer science concerns the mathematical specification and development of information systems. Typically, a specification embodies a twofold constraint: the required program must

conform to such-and-such a functional specification and terminate. Without the last conjunct, undesirable programs which execute forever, never yielding a result, might be allowed (in some models of computation). However, such a specification is no more than a behaviour phrased in terms of observables for input, output (appearing in the functional specification) and termination (appearing in the second conjunct). And we have just seen that termination cannot be assumed to be decidable.

The consequence of allowing an observable to be undecidable is that some ingenuity is required to prove that an implementation meets a specification phrased in terms of its observables: no program can possibly achieve that task in general.

4. The philosophy of the method of abstraction

The time has come to provide further conceptual clarification concerning the nature and consequences of the method of abstraction. In this section we relate the relevant work of Marr, Pylyshyn, Dennett and Davidson to the method of abstraction, and discuss the thorny issues of relativism and antirealism. A word of warning may be in order. When confronted with a new theory or method, it is natural to compare it and perhaps (mistakenly) identify it with something old and well-established. In particular, previous theories or methods can work as powerful magnets that end by attracting anything that comes close to their space of influence, blurring differences. So this section aims at putting some distance between some old acquaintances and our new proposal.

4.1. Levels of organization and of explanation

Several important ways have been proposed for speaking of levels of analysis of a system. The following two families can be singled out as most representative:

1) Levels of organization (LoOs) support an ontological approach, according to which the system under analysis is supposed to have a (usually hierarchical) structure in itself, or *de re*, which is allegedly uncovered by its description and objectively formulated in some neutral observation language (Newell [1990], Simon [1996]). For example, levels of communication, of decision processing (Mesarovic et al. [1970]) and of information flow can all be presented as specific instances of analysis in terms of LoOs.

There is a twofold connection between LoOs and LoAs. If the hierarchical structure of the system itself is thought of as a GoA, then for each constituent LoA there is a corresponding LoO. Alternatively, one can conceive the analysis of the system, not the system itself, as being the object of study. Then the method of abstraction leads one to consider a GoA whose constituent LoAs are the LoOs. Note that, since the system under analysis may be an artefact, knowledge of its LoO may be available in terms of knowledge of its specifications.

2) Levels of explanation (LoEs) support an epistemological approach, quite common in cognitive and computer science (Benjamin et al. [1998]). Strictly speaking, the LoEs do not really pertain to the system or its model. They provide a way to distinguish between different epistemic approaches and goals, such as when we analyse an exam question from the students' or the teacher's perspectives, or the description of the functions of a technological artefact from the expert's or the layperson's point of view.

A LoE is an important kind of LoA. It is pragmatic and makes no pretence of reflecting an ultimate description of the system. It has been defined with a specific practical view or use in mind. Manuals, pitched at the inexpert user, indicating "how to" with no idea of "why", provide a good example.

The two kinds of "structured analysis" just introduced are of course interrelated. Different LoEs – e.g., the end-user's LoE of how an applications package is to be used versus the programmer's LoE of how it is executed by machine – are connected with different LoAs – e.g., the end-user's LoA represented by a specific graphic interface versus the programmer's code – which in turn are connected with different LoO – e.g., the commonsensical WYSIWYG versus the software architecture. However, LoAs provide a foundation for both, and LoOs, LoEs and LoAs should not be confused. Let us consider some clarifying examples.

One of the most interesting and influential cases of multi-layered analysis is provided by Marr's three-levels hypothesis. After Marr [1982], it has become common in cognitive and philosophical studies (McClamrock [1991]) to assume that a reasonably complex system can be understood only by distinguishing between levels of analysis.

Here is how Marr himself put it: "Almost never can a complex system of any kind be understood as a simple extrapolation from the properties of its elementary

components. Consider for example, some gas in a bottle. A description of thermodynamic effects – temperature, pressure, density, and the relationships among these factors – is not formulated by using a large set of equations, one for each of the particles involved. Such effects are described at their own level, that of an enormous collection of particles; the effort is to show that in principle the microscopic and the macroscopic descriptions are consistent with one another. If one hopes to achieve a full understanding of a system as complicated as a nervous system, a developing embryo, a set of metabolic pathways, a bottle of gas, or even a large computer program, then one must be prepared to contemplate different kinds of explanation at different levels of description that are linked, at least in principle, into a cohesive whole, even if linking the levels in complete detail is impractical. For the specific case of a system that solves an information-processing problem, there are in addition the twin strands of process and representation, and both these ideas need some discussion.” (Marr [1982], pp. 19–20).

In particular, in the case of an information-processing system, Marr and his followers suggest the adoption of three levels of analysis (all the following quotations are from Marr [1982]):

1) *the computational level*. This is a description of “the abstract computational theory of the device, in which the performance of the device is characterised as a mapping from one kind of information structures, the abstract properties of this mapping are defined precisely, and its appropriateness and adequacy for the task at hand are demonstrated” (p. 24);

2) *the algorithmic level*. This is a description of “the choice of representation for the input and output and the algorithm to be used to transform one into the other” (p. 24-25);

3) *the implementational level*. This is a description of “the details of how the algorithm and representation are realized physically – the detailed computer architecture, so to speak.” (p. 25).

The three levels are supposed to be loosely connected and in a one-to-many mapping relation: for any computational description of a particular information-processing problem there may be several algorithms for solving that problem, and any algorithm may be implemented in several ways.

Along similar lines, Pylyshyn [1984] has spoken of the semantic, the syntactic, and the physical levels of description of an information-processing system, with the (level of) functional architecture of the system playing the role of a bridge between Marr's algorithmic and implementational levels. And Dennett [1987] has proposed a hierarchical model of explanation based on three different "stances": the intentional stance, according to which the system is treated as if it were a rational, thinking agent attempting to carry out a particular task successfully; the design stance, which concerns the general principles governing the design of any system that might carry out those tasks successfully; and the physical stance, which considers how a system implementing the appropriate design-level principles might be physically constructed.

The tripartite approaches of Marr, Pylyshyn and Dennett share three important features. First, they are each readily formalised in terms of GoAs with three LoAs. Second, they do not distinguish between LoO, LoE and LoA; and this because (third feature) they assign a privileged role to explanations. As a result, their ontological commitment is embedded and hence concealed. The common reasoning seems to be the following: "this is the right level of analysis because that is the right LoO", where no justification is offered for why that LoO is chosen as the right one. Nor is the epistemological commitment made explicit or defended; it is merely presupposed. This is where the method of abstraction provides a significant advantage. By starting from a clear endorsement of each specific LoA, a strong and conscious effort can be made to uncover the ontological commitment of a theory (and hence of a set of explanations), which now needs an explicit acceptance on the part of the user, and requires no hidden epistemological commitment, which now can vary depending on goals and requirements.

4.2. Conceptual schemes

The resemblance between LoAs and conceptual schemes (CSs) is close enough to require clarification. In this section, we shall briefly compare the two. The aim is not to provide an exegetical interpretation or a philosophical analysis of Davidson's famous criticism of the possibility of irreducible CSs, but rather to clarify further the nature of LoAs and

explain why LoAs can be irreducible, although in a sense different from that liked by supporters of the irreducibility of CSs.¹⁶

According to Davidson, all CSs share four features (the following quotations are from Davidson [1974]):

1) CSs are clusters or networks of (possibly acquired) categories. “Conceptual schemes, we are told, are ways of organizing experience; they are systems of categories that give form to the data of sensation; they are points of view from which individuals, cultures, or periods survey the passing scene” (p. 183).

2) CSs describe or organise the world or its experience for communities of speakers. “Conceptual schemes (languages) either organize something, or they fit it”, and as “for the entities that get organized, or which the scheme must fit, I think again we may detect two main ideas: either it is reality (the universe, the world, nature), or it is experience (the passing show, surface irritations, sensory promptings, sense-data, the given)” (p. 192).

3) CSs are inescapable, in the sense that communities of speakers are entrapped within their CSs.

4) CSs are not intertranslatable.

Davidson argues against the existence of CSs as inescapable (from within) and impenetrable (from without) ways of looking at the world by interpreting CSs linguistically and then by trying to show that feature (4) is untenable. Could the strategy be exported to contrast the existence of equally inescapable and impenetrable LoAs? Not quite.

Let us examine what happens to the four features above when LoAs are in question:

a) LoAs are clusters or networks of observables. Since they deal with observables, LoAs are not an anthropocentric prerogative but allow a more general (or indeed less biased) approach. We do not have to limit ourselves to human beings or to communities of speakers. Different sorts of empirical or abstract agents – not only human beings but also computers, animals, plants, scientific theories, measurement instruments etc. – operate and deal with the world (or, better, with the data they glean from it) at some LoAs. By

¹⁶ Newell reached similar conclusions, despite the fact that he treated LoA as LoO, an ontological form of levellism that allowed him to escape relativism and antirealism more easily, see Newell [1982] and Newell [1993].

neatly decoupling LoAs from the agents that implement or use them, we avoid the confusion between CSs, the languages in which they are formulated or embodied, and the agents that use them. We shall return to this point presently.

b) LoAs model the world or its experience. LoAs are anchored to their data, in the sense that they are constrained by them; they do not describe or organise them, they actually build models out of them. So the relation between models and their references (the analysed systems) is neither one of discovery, as in Davidson's CSs, nor one of invention, but one of design, to use an equally general category. It follows that, contrary to Davidson's CSs, it makes no sense to speak of LoAs as Xerox machines or personal organisers of some commonly shared ontology (the world or its experience). Ontological commitments are initially negotiated through the choice and shaping of LoAs, which therefore cannot presuppose a metaphysical omniscience.

Because of the differences between (1)–(2) and (a)–(b), the remaining two features acquire a significantly different meaning, when speaking of LoAs. Here is how the problem is reformulated. LoAs generate, and commit the agent to, information spaces. In holding that some LoAs can be irreducible and hence untranslatable we are not arguing that:

i) agents using LoAs can never move seamlessly from one information space to another. This is false. They obviously can, at least in some cases: just imagine gradually replacing some observables in the LoAs of an agent. This is equivalent to arguing that human beings cannot learn different languages. Note, however, that some agents may have their LoAs hardwired: imagine, for example, a thermometer;

ii) agents using LoAs can never expand their information spaces. This is also false. Given the nested nature of some LoAs and the possibility of constructing supersets of sets of observables, agents can aggregate increasingly large information spaces. This is equivalent to arguing that human speakers cannot expand their languages semantically, another obvious nonsense.

So, if we are talking about the agents using or implementing the LoAs, we know that agents can sometimes modify, expand or replace their LoAs, and hence some degree of intertranslatability, understood as the acquisition or evolution of new LoAs, is

guaranteed. The point in question is another one, however, and concerns the relation between the LoAs themselves.

LoAs are the place at which (diverse) independent systems meet and act on or communicate with each other. If one reads carefully, one will notice that this is the definition of an interface. The systems interfaced may adapt or evolve their interfaces or adopt other interfaces, as in (i) and (ii), yet different interfaces may still remain mutually untranslatable. Consider, for example, the “tasting LoA” and the “purchasing LoA” in our wine example. But if two LoAs are untranslatable, it becomes perfectly reasonable to assume that:

iii) agents may inhabit only some types of information spaces in principle.

Some information spaces may remain inaccessible not just in practice but also in principle, or they may be accessible only asymmetrically, to some agents. Not only that, but given the variety of agents, what is accessible to one or some may not be accessible to all. This is easily explained in terms of modal logic and possible worlds understood as information spaces. The information space of a child is asymmetrically accessible from the information space of an adult, and the information space of a bat overlaps insufficiently with the information space of any human agent to guarantee a decent degree of translatability (Nagel [1974]).

In principle, some information spaces may remain forever disjoint from any other information spaces that some agents may be able to inhabit. When universalised, this is Kant’s view of the noumenal world, which is accessible only to its creator. Does this imply that, after all, we are able to say what a radically inaccessible information space would be like, thus contradicting ourselves? Of course not. We are only pointing in the direction of the ineffable, without grasping it. It is a bit like drawing a figure without ever being able to paint it.

To return to Davidson, even conceding that he may be successful in criticising the concept of CSs, his arguments do not affect LoAs. The problem is that Davidson limits his considerations to information spaces that he assumes, without much reason, to be already linguistically and ontologically delimited. When this is the case, one may concede his point. However, LoAs do not vouch for the kind of epistemic realism, verificationism, panlinguism and representationist view of knowledge that Davidson

implicitly assumes in analysing CSs. And once these fundamental assumptions are eliminated, Davidson's argument loses most of its strength. Incommensurable and untranslatable LoAs are perfectly possible. We shall see that this provides no good ground for a defence of some form of radical conceptual relativism (section 4.3) or anti-realism (section 4.4).

Davidson's criticism ends by shaping an optimistic approach to the problem of the incommensurability of scientific theories that we cannot share, but then, what conclusions can be drawn, from our analysis of LoAs, about the anti-realist reading of the history of science? An unqualified answer would fall victim to the same fallacy of un-layered abstraction we have been denouncing in the previous pages. The unexciting truth is that different episodes in the history of science are more or less comparable depending on the LoA adopted. Consider the great variety of building materials, requirements, conditions, needs and so on, which determine the actual features of a building. Does it make sense to compare a ranch house, a colonial home, a town house, a detached house, a semidetached house, a terraced house, a cottage, a thatched cottage, a country cottage, a flat in a single-storey building, and a Tuscan villa? The question cannot be sensibly answered unless one specifies the LoA at which the comparison is to be conducted. Likewise, our answer concerning the reading of the history of science is: given the nature of LoAs, it is always possible to formulate a LoA at which comparing different episodes in the history of science makes perfect sense. But do not ask absolute questions, for they just create an absolute mess.

4.3. Pluralism without relativism

A LoA qualifies the level at which a system is considered. In this paper, we have argued that it must be made clear before the properties of the system can sensibly be discussed. In general, it seems that many disagreements might be clarified and resolved if the various "parties" make explicit their LoA. By structuring the explanandum, LoAs can reconcile the explanans. Yet, another crucial clarification is now in order. It must be stressed that a clear indication of the LoA at which a system is being analysed allows pluralism without falling into relativism or "perspectivism", a term coined by Hales and Welshon [2000] in connection with Nietzsche's philosophy. As remarked above, it would

be a mistake to think that “anything goes” as long as one makes the LoA explicit, because LoAs can be mutually comparable and assessable, in terms of inter-LoA coherence, of their capacity to take full advantage of the same data and of their degree of fulfilment of the explanatory and predictive requirements laid down by the level of explanation. Thus, introducing an explicit reference to the LoA makes it clear that the model of a system is a function of the available observables, and that it is reasonable to rank different LoAs and to compare and assess the corresponding models.

4.4. Realism without descriptivism

For a typed variable to be an observable it must be interpreted, a correspondence that has inevitably been left informal. This interpretation cannot be omitted: a LoA composed of typed variables called simply *x*, *y*, *z* and so on and treated rather formally, would leave the reader (or the writer some time later) with no hint of its domain of application. Whilst that is the benefit of mathematics, enabling its results to be applied whenever its axioms hold, in the method of abstraction it confers only obscurity. Does the informality of such interpretation hint at some hidden circularity or infinite regress? Given the distinction between LoO and LoA, and the fact that there is no immediate access to any LoO that is LoA-free, how can an observable be defined as “realistic”? That is, must the system under consideration already be observed before a “realistic” observation can be defined? The mathematics underlying our definitions of typed variable and behaviour has been indicated (even if it is not always fully used in practice) to make the point that, in principle, the ingredients in a LoA can be formalised. There is no circularity: the heuristically appreciated system being modelled never exists on the same plane as that being studied methodically.

The point might be clarified by considering Tarski’s well-known model-theoretic definition of truth (Tarski [1944]). Is there circularity or regress there? Might it be argued that one needs to know truth before defining it, as Meno would have put it? Of course not, and the same resolution is offered here. Tarski’s recursive definition of truth over syntactic construction is based on an appreciation of the properties truth is deemed to have, but that appreciation and the rigorous definition exist on “different planes”. So circularity is avoided.

More interesting is the question of infinite regress. Tarski's definition formalises certain specific properties of truth; a regress would obtain only were a complete characterisation sought. So it is with the interpretation required to define an observable. Some property of an undisclosed system is being posited at a certain level of abstraction. An unending sequence of LoAs could possibly obtain were a complete characterisation of a system sought.

It is implicit in the method of abstraction that a GoA is to be chosen that is accurate or "realistic". How, then, is that to be determined without circularity? We give the answer traditionally offered in mathematics and in science: it is determined by external adequacy and internal coherence or, in computer jargon, validation (the GoA satisfies its operational goals) and verification (each step in the development of the GoA satisfies the requirements imposed by previous steps). First, the behaviours at a moderated LoA must adequately reflect the phenomena sought by complying with their constraints; if not, then either the definition of the behaviour is wrong or the choice of observables is inappropriate. When the definition of observables must incorporate some "data", the latter behave like constraining affordances and so limit the possible models. We refer to Floridi [2004] for further details and examples. Second, the condition embodied in the definition of GoA is a remarkably strong one, and ensures a robust degree of internal coherence between the constituent LoAs. The multiple LoAs of a GoA can be thought of as interlocking like the answers to a multidimensional cross-word puzzle. Though such consistency does not guarantee that one's answer to the cross-word is the same as the originator's, it drastically limits the number of solutions, making each more likely.

Adequacy/validation and coherence/verification neither entail nor support naive realism. GoAs ultimately construct models of systems. They do not describe or portray or uncover the intrinsic nature of the systems they analyse. We understand systems derivatively, only insofar as we understand their models. Adequacy and coherence are the most we can hope for.

5. Conclusion

Feynman once remarked that “if we look at a glass of wine closely enough we see the entire universe. [...] If our small minds, for some convenience, divide this glass of wine, this universe, into parts – physics, biology, geology, astronomy, psychology, and so on – remember that nature does not know it!”.¹⁷ In this paper, we have shown how the analysis of the glass of wine may be conducted at different levels of epistemological abstraction without assuming any corresponding ontological levellism. Nature does not know about LoAs either.

In the course of the paper we have introduced the epistemological method of abstraction and applied it to the study, modelling and analysis of phenomenological and conceptual systems. We have demonstrated its principal features and main advantages. Yet one may object that, by providing a few simple examples and some tailored case-based analyses, the method really predates its applications, which were merely chosen and shaped for their suitability. In fact, it is exactly the opposite: we were forced to develop the method of abstraction when we encountered the problem of defining the nature of agents (natural, human and artificial) in Floridi and Sanders [2004]. Since then, we have been applying it to some long-standing philosophical problems in different areas. We have used it in computer ethics, to argue in favour of the minimal intrinsic value of informational objects (Floridi [2003]); in epistemology, to prove that the Gettier problem is not solvable (Floridi [forthcoming]); in the philosophy of mind, to show how an agent provided with a mind may know that she has one and hence answer Dretske’s question “how do you know you are not a zombie?” (Floridi [forthcoming]); in the philosophy of science, to propose and defend an informational approach to structural realism that reconciles forms of ontological and epistemological structural realism (Floridi [forthcoming]); and in the philosophy of AI, to provide a new model of telepresence (Floridi [forthcoming]). In each case, the method of abstraction has been shown to provide a flexible and fruitful approach. Clearly, the adoption of the method of abstraction raises interesting questions, such as why certain LoAs, e.g. the so-called “naive physics” view of the world and the “folk psychology” approach to the mind,

¹⁷ Feynman [1995], the citation is from the Penguin edition, p. 66.

appear to be “privileged”, or whether artificial life (ALife) can be defined in terms of GoA. So much work lies ahead.

The method clarifies implicit assumptions, facilitates comparisons, enhances rigour and hence promotes the resolution of possible conceptual confusions. It also provides a detailed and controlled way of comparing analyses and models. Yet, all this should not be confused with some neo-Leibnizian dream of a “calcuemus” approach to philosophical problems. Elsewhere (Floridi [2004]), we have argued that genuine philosophical problems are intrinsically *open*, that is, they are problems capable of different and possibly irreconcilable solutions, which allow honest, informed and reasonable differences of opinion. The method we have outlined seeks to promote explicit solutions, which facilitate a critical approach and hence empower the interlocutor. It does not herald any sort of conceptual “mechanics”.

The method is not a panacea either. We have argued that, for discrete systems, whose observables take on only finitely-many values, the method is indispensable. Nevertheless, its limitations are those of any typed theory. Use of LoAs is effective in precisely those situations where a typed theory would be effective, at least informally. Can a complex system always be approximated more accurately at finer and finer levels of abstraction, or are there systems which can simply not be studied in this way? We do not know. Perhaps one may argue that the mind or society – to name only two typical examples – are not susceptible to such an approach. In this paper we have made no attempt to resolve this issue.

We have also avoided committing ourselves to determining whether the method of abstraction may be exported to ontological or methodological contexts. Rather, we have defended a version of epistemological levellism that is perfectly compatible with the criticisms directed at other forms of levellism.

Introduction of LoAs is often an important step prior to mathematical modelling of the phenomenon under consideration. However, even when that further step is not taken, introduction of LoAs remains a crucial tool in conceptual analysis. Of course, care must be exercised in type-free systems, where the use of the method may be problematic. Such systems are susceptible to the usual paradoxes and hence to inconsistencies, not only when formalised mathematically but also when considered informally. Examples of

such systems arise frequently in philosophy and in artificial intelligence. However, we hope to have shown that, if carefully applied, the method confers remarkable advantages in terms of careful treatment, consistency and clarity.

Acknowledgements

This is a fully revised version of a paper given at a philosophy graduate seminar in Bari. We are grateful to the students for their feedback. We also wish to thank Gian Maria Greco, Gianluca Paronitti and Matteo Turilli for their discussions of several previous drafts, Paul Oldfield for his editorial suggestions and Carl Craver for having made available to us his forthcoming research.

References

- Arbib, M. A. 1989, *The Metaphorical Brain II: Neural Networks and Beyond* (New York: John Wiley & Sons).
- Barwise, J., and Etchemendy, J. 1987, *The Liar: An Essay on Truth and Circularity* (New York; Oxford: Oxford University Press).
- Bechtel, W., and Richardson, R. C. 1993, *Discovering Complexity: Decomposition and Localization as Strategies in Scientific Research* (Princeton: Princeton University Press).
- Bell, J. S. 1987, "The Theory of Local Beables" in *Speakable and Unspeakable in Quantum Mechanics - Collected Papers on Quantum Mechanics* (Cambridge: Cambridge University Press), 52-62.
- Benjamin, P., Erraguntla, M., Delen, D., and Mayer, R. 1998, "Simulation Modeling and Multiple Levels of Abstraction" in *Proceedings of the 1998 Winter Simulation Conference*, edited by D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan (Piscataway, New Jersey: IEEE Press), 391-398.
- Block, N. 1997, "Anti-Reductionism Slaps Back" in *Philosophical Perspectives 11: Mind, Causation, and World*, edited by J. E. Tomberlin (Oxford - New York: Blackwell), 107-133.
- Boolos, G., Burgess, J. P., and Jeffrey, R. C. 2002, *Computability and Logic* 4th ed. (Cambridge: Cambridge University Press).
- Brown, H. C. 1916, "Structural Levels in the Scientist's World", *The Journal of Philosophy, Psychology and Scientific Methods*, 13(13), 337-345.
- Cariani, P. 1991, "Emergence and Artificial Life" in Langton et al. [1992], 775-797.
- Craver, C. F. 2004, "A Field Guide to Levels", *Proceedings and Addresses of the American Philosophical Association*, 77(3).
- Craver, C. F. forthcoming, *Explaining the Brain: A Mechanist's Approach*.
- Damper, R. I. 2000, "Emergence and Levels of Abstraction", *International Journal of Systems Science*, 31(7), 811-818.
- Davidson, D. 1974, "On the Very Idea of a Conceptual Scheme", *Proceedings and Addresses of the American Philosophical Association*, 47. Reprinted in *Inquiries into Truth and Representation* (Oxford: Clarendon Press, 1984): 183-98. All page

- numbers to the quotations in the text refer to the reprinted version.
- de Roeper, W.-P., and Engelhardt, K. 1998, *Data Refinement: Model-Oriented Proof Methods and Their Comparison* (Cambridge: Cambridge University Press).
- Dennett, D. C. 1971, "Intentional Systems", *The Journal of Philosophy*, (68), 87-106.
- Dennett, D. C. 1987, *The Intentional Stance* (Cambridge, Mass; London: MIT Press).
- Egyed, A., and Medvidovic, N. 2000, "A Formal Approach to Heterogeneous Software Modeling" in *Proceedings of the Third International Conference on the Fundamental Approaches to Software Engineering (Fase 2000, Berlin, Germany, March-April) - Lecture Notes in Computer Science, No. 1783*, edited by Tom Mailbaum (Berlin/Heidelberg: Springer-Verlag),
- Emmeche, C., Køppe, S., and Stjernfelt, F. 1997, "Explaining Emergence: Towards an Ontology of Levels", *Journal for General Philosophy of Science*, 28, 83-119.
- Feynman, R. P. 1995, *Six Easy Pieces* (Boston, MA.: Addison-Wesley).
- Floridi, L. 2003, "On the Intrinsic Value of Information Objects and the Infosphere", *Ethics and Information Technology*, 4(4), 287-304.
- Floridi, L. 2004, "Information" in *The Blackwell Guide to the Philosophy of Computing and Information*, edited by L. Floridi (Oxford - New York: Blackwell), 40-61.
- Floridi, L. 2004, "Open Problems in the Philosophy of Information", *Metaphilosophy*, 35(4), 554-582.
- Floridi, L. forthcoming, "Consciousness, Agents and the Knowledge Game".
- Floridi, L. forthcoming, "The Informational Approach to Structural Realism".
- Floridi, L. forthcoming, "On the Logical Unsolvability of the Gettier Problem", *Synthese*.
- Floridi, L. forthcoming, "Presence: From Epistemic Failure to Successful Observability", *Presence: Teleoperators and Virtual Environments*.
- Floridi, L., and Sanders, J. W. 2004, "On the Morality of Artificial Agents", *Minds and Machines*, 14(3), 349-379.
- Foster, C. L. 1992, *Algorithms, Abstraction and Implementation: Levels of Detail in Cognitive Science* (London: Academic Press).
- Gell-Mann, M. 1994, *The Quark and the Jaguar: Adventures in the Simple and the Complex* (London: Little Brown).
- Hales, S. D., and Welshon, R. 2000, *Nietzsche's Perspectivism* (Urbana: University of

- Illinois Press).
- Hayes, I., and Flinn, B. 1993, *Specification Case Studies* 2nd ed (New York; London: Prentice Hall).
- Heil, J. 2003, "Levels of Reality", *Ratio*, 16(3), 205-221.
- Hendriks-Jansen, H. 1989, "In Praise of Interactive Emergence: Or Why Explanations Don't Have to Wait for Implementations" in Langton *et al.* [1989], 282-299.
- Hoare, C. A. R., and He, J. 1998, *Unifying Theories of Programming* (London: Prentice Hall).
- Hughes, P., and Brecht, G. 1976, *Vicious Circles and Infinity: A Panoply of Paradoxes* (London: Cape). Originally published: Garden City, N.Y.: Doubleday, 1975.
- Kelso, J. A. S. 1995, *Dynamic Patterns: The Self-Organization of Brain and Behavior* (Cambridge, Mass; London: MIT Press).
- Knuth, D. E. 1997, *The Art of Computer Programming* 3rd ed. (Reading, Mass.; Harlow: Addison-Wesley). 3 vols.
- Marr, D. 1982, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (San Francisco: W.H. Freeman).
- McClamrock, R. 1991, "Marr's Three Levels: A Re-Evaluation", *Minds and Machines*, 1, 185-196.
- Mesarovic, M. D., Macko, D., and Takahara, Y. 1970, *Theory of Hierarchical, Multilevel, Systems* (New York: Academic Press).
- Mitchell, T. M. 1997, *Machine Learning* International edition (New York; London: McGraw-Hill).
- Moor, J. H. (ed.) 2001, *The Turing Test: Past, Present and Future* (Special issue of *Minds and Machines*, 11(1)).
- Nagel, T. 1974, "What Is It Like to Be a Bat?" *The Philosophical Review*, 83(4), 435-450.
- Newell, A. 1982, "The Knowledge Level", *Artificial Intelligence*, 18, 87-127.
- Newell, A. 1990, *Unified Theories of Cognition* (Cambridge, Mass; London: Harvard University Press).
- Newell, A. 1993, "Reflections on the Knowledge Level", *Artificial Intelligence*, 59, 31-38.
- Oppenheim, P., and Putnam, H. 1958, "The Unity of Science as a Working Hypothesis"

- in *Minnesota Studies in the Philosophy of Science. Concepts, Theories, and the Mind-Body Problem.*, edited by H. Feigl, Michael Scriven, and Grover Maxwell (Minneapolis: University of Minnesota Press), vol. 2, 3-36.
- Poli, R. 2001, "The Basic Problem of the Theory of Levels of Reality", *Axiomathes*, 12, 261–283.
- Pylyshyn, Z. W. 1984, *Computation and Cognition: Toward a Foundation for Cognitive Science* (Cambridge, Mass: MIT Press).
- Robinson, J. 1989, *Vintage Timecharts: The Pedigree and Performance of Fine Wines to the Year 2000* (London: Mitchell Beazley).
- Robinson, J. (ed.) 1994, *The Oxford Companion to Wine* (Oxford: Oxford University Press).
- Russell, B. 1902, "Letter to Frege" In *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*, ed. by J. van Heijenoort (Harvard University Press: Cambridge, MA, 1967), 124-125.
- Salthe, S. N. 1985, *Evolving Hierarchical Systems: Their Structure and Representation* (New York: Columbia University Press).
- Schaffer, J. 2003, "Is There a Fundamental Level?" *Nous*, 37(3), 498-517.
- Shieber, S. 2004, *The Turing Test - Verbal Behavior as the Hallmark of Intelligence* (Cambridge, Mass.: MIT).
- Simon, H. A. 1969, *The Sciences of the Artificial* 1st ed. (Cambridge, Mass. - London: MIT Press). The text was based on the Karl Taylor Compton lectures, 1968.
- Simon, H. A. 1996, *The Sciences of the Artificial* 3rd ed. (Cambridge, Mass.; London: MIT Press).
- Spivey, J. M. 1992, *The Z Notation: A Reference Manual* 2nd ed (New York; London: Prentice-Hall).
- Tarski, A. 1944, "The Semantic Conception of Truth and the Foundations of Semantics", *Philosophy and Phenomenological Research*, 4, 341-376. Reprinted in L. Linsky (ed.) *Semantics and the Philosophy of Language* (Urbana: University of Illinois Press, 1952).
- Turing, A. M. 1950, "Computing Machinery and Intelligence", *Minds and Machines*, 59, 433-460.

Wimsatt, W. C. 1976, "Reductionism, Levels of Organization and the Mind-Body Problem" in *Consciousness and the Brain*, edited by G. Globus, G. Maxwell, and I. Savodnik (New York: Plenum), 199-267.