What Can We Know of Computational Inforgs?

Federico Gobbo
DISIM – Dep. of Inf. Eng.,
Comp. Sc., and Mathematics
University of L'Aquila, Italy
federico.gobbo@univaq.it

Marco Benini*
Dept. of Pure Mathematics
School of Mathematics
University of Leeds, UK
M.Benini@leeds.ac.uk

Manuscript submitted to the Fifth Workshop on the Philosophy of Information, 2013

Abstract

Within the ontology of Informational Structural Realism, an informational organism (inforg) carries a minimal ontological commitment in favour of structural properties of reality, which answers the question 'what can we know?' (Floridi, 2011, ch. 15). In this paper, we deal with the answer to this question in the case of computational inforgs (c-inforgs), whose engineered artifact are based on some kind of computing device, typically a Von Neumann machine (VNM) (Gobbo and Benini, 2013, for details).

The system-level-model-structure (SLMS) scheme relies on the method of levels of abstraction (LoAs), where a LoA is individuated according to the range of observables analysed within a system, then producing a model that identifies a structure (Floridi, 2011, 15.2.3). When applying the SLMS scheme to the case study of c-inforgs, we should distinguish between two different classes.

The first class is populated by *open* c-inforgs. Members of this class show an important property in their VNM-based counterpart: the openness of the source code—a fundamental Level of Organisation (LoO) in c-inforgs, see Gobbo and Benini (forthcoming). Openness can be loosely defined as the ability to inspect the VNM-based counterpart of the c-inforg to anyone—extending the usual definition of availability of the source code to anybody without barriers. Thus, we may have *direct* knowledge of the artifact under investigation, although directness does not imply that knowledge is transparent, because it is mediated by the appropriate LoA(s). Following the well-known metaphor of boxes, open c-inforgs are grey to some degree.

The second class is the dual to the first one, being populated by *closed* c-inforgs. Here, the access to the VNM counterpart, especially the source code, is not granted. However, we can still access inputs and output: even if information got hidden into a black box, the logical structure of the VNM-based artifact is still open to inspection. As put effectively by Franklin (1999, 721): 'despite the best efforts of Microsoft to make all computer programs so large as to be incomprehensible, small surveyable programs are still common items'. In the case of closed c-inforgs we can only infer *indirectly* their structural properties from the observation of how behaviours change according to inputs, which is an eminently qualitative activity. According to the method of LoAs, in the case of closed c-inforgs we can only observe the feedback given by

*Dr Benini was supported by a Marie Curie Intra European Fellowship, grant n. PIEF-GA-2010-271926, *Predicative Theories and Grothendieck Toposes*, within the 7th European Community Framework Programme.

the system: the model generated in the SLMS scheme is at the 2^{nd} order, in terms of ontological commitment (Floridi, 2011, 15.2.4).

Evidently, the analyses of open c-inforgs can be more fine-grained than the ones made on closed c-inforgs: openness let the analysts use quantitative methods to measure the VNM counterpart, in particular, the source code, which can be inspected using 'direct metrics' coming from Software Engineering. Direct metrics do not depend upon a measure of any other attribute, and therefore they are presumed valid *per se*, according to IEEE Standard 1061 (IEEE, 1998). It is important to notice that direct metrics are necessary but not sufficient, and so other metrics have been proposed and validated in terms of the first ones: it is not by chance that the IEEE Standard 1061 presents the methodology for software *quality* metrics. We can now instantiate the SLMS scheme of software quality metrics: the ontological commitment at the 1^{st} order is represented by LoAs representing direct metrics, which are inherently quantitative; on the other hand, indirect metrics are more qualitative-oriented LoAs, as they strictly depend on the first ones by definition, and so they are at the 2^{nd} order in terms of ontological commitment.

Software Engineering can give us a concrete field of application. Let us take a complex information system as a closed c-inforg: the VNM-based artifact is made by the complex of hardware, software (source and running code), and network infrastructure, while its biological counterpart is made by software programmers, system administrators, end-users etc.. It is well known in the literature that even a focused goal-oriented engineering practice like the risk assessment of a given complex information system is 'a subjective process' (Redmill, 2002), as it depends on the subjective views of the expert. Nevertheless, a formalisation of the risk assessment procedure was recently proposed, introducing the notion of compatible metrics in terms of morphisms between partial orders (Benini and Sicari, 2009).

A future direction of work is to generalise this formalisation originally found in Software Engineering to compare *qualitative* properties of c-inforgs. For example, Facebook is a very complex c-inforg, which is closed to its end-users. We consider the relation between an end-user (the biological agent) and his/her account in the social network (the VNM-based counterpart considered here). We notice that some users prefer to access their Facebook account through a smartphone, while other prefer their favourite web browser of their own laptop. Now, the possible actions a user can perform on Facebook through a smartphone are not identical to the actions available through the laptop web browser—for instance, options to share one's posts, or the presence of a photocamera. As a result, we can describe two different LoAs (in terms of graphic end-user's interface), two corresponding LoOs (the mobile application of Facebook is an application software, while the web site for standard web browser is another one) and finally two corresponding Levels of Explanation (LoEs), expressed in terms of goals or users' habits, e.g., users keen to smartphones could prefer to give and receive GPS-based information, while users keen to desktops perhaps prefer to write longer texts in posts. Order theory may be used to formalise subjective judgements like 'x is better than y' or 'A prefers z to that w'. Our aim is eventually to describe a user's preference profile in terms of partial orders, so that we can algebraically measure the distance between the propensities of different biological users within the same c-inforg (in our example, end-users in Facebook).

In conclusion, the test case of c-inforgs shows us there is always the mediation of at least one LoA, which presents at least a minimal degree of quality even if expressed in quantitative terms (liminal realism). Therefore, the working philosopher of information cannot avoid the continuous interplay between qualitative and quantitative point of views of information, especially after the computational turn, when informational organisms become more and more complex.

References

Benini, M. and Sicari, S. (2007), 'Risk assessment via partial orders', *Advances in Computer Science and Engineering* **3**(1), 19–46.

Floridi, L. (2011), *The Philosophy of Information*, Oxford University Press, Oxford.

Franklin, J. (1999), 'Structure and domain-independence in the formal sciences', *Stud. Hist. Phil. Sci.* **30**(4), 721–723.

Gobbo, F. and Benini, M. (2013), 'The Minimal Levels of Abstraction in the History of Modern Computing', *Philosophy & Technology* pp. 1–17.
URL: *http://dx.doi.org/10.1007/s13347-012-0097-0*

Gobbo, F. and Benini, M. (forthcoming), 'Why zombies can't write significant source code: The knowledge game and the art of computer programming', *Journal of Experimental & Theoretical Artificial Intelligence* .

IEEE (1998), 'Standard for a software quality metrics methodology', IEEE Standards Dept. Std. 1061-1998.

Redmill, F. (2002), 'Risk analysis: A subjective process', *Engineering Management Journal* **12**(2), 91–96.