The aim of this presentation is to show that computational phenomena are also intentional phenomena and that this is particularly manifest in agent-based social simulation. The first step of our presentation is to introduce some epistemological perspectives of computer science in the epistemology of social simulation, especially in the epistemology of agent-based social simulation. The epistemological status of simulation cannot be studied adequately without the contribution of the epistemology of computer science. In particular it is important to know the kind of thing that computer programs are supposed to be, and how knowledge of program executions in computers might be acquired.

These problems are related to a classic debate in computer science, which confronts researchers advocating the use of formal methods for verifying programs with those advocating the use of empirical methods. Particularly, it is related to the need of distinguishing between mathematical theorems, scientific theories and computer programs.

<table>
<thead>
<tr>
<th></th>
<th>Mathematical Proofs</th>
<th>Scientific Theories</th>
<th>Computer Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic Entities</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Semantic Significance</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Causal Capability</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

Table 1 – Proofs, scientific theories and computer programs.

Computer programs, like scientific theories, have semantical significance that mathematical proofs do not possess. However, scientific theories do not possess the causal capabilities of computer programs, which can affect the performance of computers when they are loaded and executed. In relation to this presentation, the main premise of computer science epistemology is that programs, unlike scientific theories, seem to possess a causal capability that scientific
theories do not seem to possess. Since we do not have much space in this extended abstract, we refer to Fetzer (2001).

In addition, the epistemological status of computer simulation cannot be assessed without a preliminary identification of its various research goals. The variety of research goals in agent-based social simulation can be characterised according to several paradigmatic models used in the field, which seem to call for different methodologies of research. Figure 1 illustrates a classification of paradigmatic models in agent-based social simulation, as presented in David et al (2004).

![Figure 1 – A classification of paradigmatic models in Agent-Based Social Simulation.](image)

In the context of this workshop we are interested in the left part of the figure, which involves social-scientific models. In that case, the role of simulation is to construct agent-based models based on the scientific corpus of the Social Sciences, in order to simulate theories or social phenomena. For further details see David et al. (2004).

The second step of our presentation is to analyse the role of computer programs in the specific context of social-scientific models. In particular, we are interested in the kind of knowledge that can be known about those computer programs, and how the knowledge of program executions might be acquired. The first thing that should be borne in mind is that the meaning of the terms verification and validation in computer science is different from the meaning that is usually ascribed to those terms in the social sciences. Nevertheless, both terms are used in social simulation and with disparate meanings (for instance, confront Gilbert and Troitzsch, 1999, pp.21-24, or David et al., 2003, with Axelrod, 1997).

If one adopts the computer science terminology, the methodology of research in simulation can be expressed according to two steps. The first step, verification, tries to find out empirical evidence about the relationship between two or more symbolic representations and the
behaviour of the computer. The second step, validation, tries to find out evidence about the relationship between those representations and the particular world that they represent.

Figure 2 – Interpretative references in simulation.

In figure 2 we put forward a simplified loop, which illustrates the interdependence between verification and validation. In a first stage there is a theory T0, interpreted according to some target of interest. Subsequently, there are two steps:

(i) Program verification: the theory T0 must be ultimately stated in terms of a set of computational programs, which are also symbolic representations. These symbolic representations are tentatively implemented in a physical computer (the hardware), with the help of other programs already implemented and running in the computer, such as a simulation environment. The simulation happens through the execution of the program in a controlled way, once again with the help of other programs already running in the computer. It is the interaction between symbolic and physical processes that leads to a causal effect of the theory T0 in the simulation behaviour. Insofar as we cannot know a priori if the observed behaviour of the simulation will correspond to the theory T0, we must inquire the behaviour of the simulation empirically. Thus, we construct a new theory T1 about the behaviour of the simulation, which should be identical to theory T0. Hence, we should try to accept or refute theory T1, according to what we believe how the computer should behave by simulating T0. If theory T1 is always refuted, this means that the tentative implementation of T0 may have been inadequate, but the theory T0 itself may be not. In that case we assume that theory T0 is methodologically “unfalsifiable”, meaning that it should neither be accepted nor refuted. Since T0 is methodologically unfalsifiable, we must find what happened during the translation of theory T0 to the implementation of T0. But once we decide to accept theory T1,
we should try to validate both theories T0 and T1, according to the logic of step (ii) as follows.

(ii) Validation of theories: in a second stage we have theories T0, T1 and their relationships to the target. Usually, the theory T1 supersedes or should be more informative than theory T0, insofar as we analyse T1 in order to find valid consequences of T0. In principle, for any consequence that is inferred formally from T0, or validated against the target, then the same conclusion must be reached when we use T1. But since theory T1 may include statements that were not considered in T0, we may not be able to infer those statements from T0. Nevertheless, if we find that T1 is not valid against the target this does not necessarily refute T0, since T1 is only a theory about the behaviour of the simulation. Before refuting theory T0, we must do further empirical inquiry for analysing the behaviour of the simulation, according to the logic of step (i). Indeed, unlike theory T0, which is always a theory about the target, the theory T1 may be interpreted according to the simulation or the target, that is to say, two different models for the same theory.

According to the above description, and from a basic premise in the Theory of Computation\(^1\), the goal of our presentation in the workshop is to reinforce the idea that computation in social simulation is also an intentional phenomenon (David, 2004). In short we want to show the following:

(i) since it is widely recognised that computation is in one way or another a symbolic or a representational or a semantical phenomenon, it is intentional insofar as we assume that the states of a computer can simulate or represent other things in the world;

(ii) the recognition that computer programs possess a causal capability which affects the performance of computers and that the researcher has the intention of submitting a set of computer behaviours that mean other things in the world to a community of observers, that should accept or reject those particular scenarios.

(iii) the recognition that the observers and researchers’ intended meanings will remain intentional, insofar as the propositions used to interpret the computers behaviour are not verified empirically.

Since models in agent-based social simulation are often specified informally, and the expressiveness of the specification languages cannot be captured by a first-order language, then the kind of knowledge that can be known about computer programs in social simulation should not be considered empirical. Programs in social simulation do not only possess a causal capability, but also an intentional capability, which in classic computer science

\(^1\) There will always be a first-order language that can simulate any execution of a program that terminates.
programs do not seem to possess. The logic of social simulation contrasts the existence of two kinds of program verification: *empirical* and *intentional* verification of programs. The difference between empirical and intentional verification of programs reflects a distinction in the kind of knowledge that can be known about social simulations.

<table>
<thead>
<tr>
<th>Scientific Theories</th>
<th>Computer Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic Entities</td>
<td>yes</td>
</tr>
<tr>
<td>Semantic Significance</td>
<td>yes</td>
</tr>
<tr>
<td>Causal Capability</td>
<td>não</td>
</tr>
<tr>
<td>Intentional Capability</td>
<td>não</td>
</tr>
</tbody>
</table>

Table 2 – Scientific theories and programs (David, 2004).

Finally, the third step of our presentation is to characterise the logic of intentional verification of programs, and characterise the logic of scientific research in social simulation accordingly. The intentional aspect of computation means that simulations cannot be characterised as theories themselves. Agent-based social simulations are a way of constructing new objects in the world. They use implementations as tools of theory building.


David, Nuno; Marietto, Maria; Sichman, Jaime; Coelho, Helder (2004). “The Structure and Logic of Interdisciplinary Research in Agent-Based Social Simulation”. In *Journal of Artificial Societies and Social Simulation* (JASSS), v.7, n.3, http://www.soc.surrey.ac.uk/JASSS/7/3/4.html

