Guidelines

> “*If a problem has no solution, it may not be a problem, but a fact—not to be solved, but to be coped with over time.*”
>
> Shimon Peres

Disclaimer

We are very aware that our ambitions for the book are high and that we are asking for a lot. There is enough time and we are willing to offer more help than one might expect from editors of typical collected volumes. We hope that you believe that this kind of collaborative work is achievable and needed, even though the end-product might be but an approximation of what is aimed for (or, perhaps, by aiming too high we might actually hit the target).

General idea - recap

We want to write a book that focuses on the question “What is a (computer) program?” -- this question is not expected to receive one definite answer. It is used as a means to stimulate deeper reflection and critical thinking and as a starting point for a pluralistic, non-reductive exploration. Rather than committing to one specific method, it is our aim to develop a work which is open-minded, a-disciplinary and collaboratively written in a way

- where no chapter is committed to one specific “discipline” but offers a variety of perspectives and uncover parallels and analogies that remain hidden when seen from just one perspective;
- which inspires an open minded reader to think about programs in other ways;
- which asks for an intellectual effort from the reader but without assuming that she has to read several other books first (see also *Level of difficulty and readership*);
- which is the start of something not the end -- we hope the book will encourage more people to do some more critical reflection.
- which sees programs as unique entities that bridge very different worlds with very different focus such as physical reality of machines, world of mathematical abstractions, practical worlds of programming and business and wider society.

Thus:

- *Be speculative, be free in your thinking.* We want to uncover parallels and analogies that remain hidden when seen from just one perspective.
- *Make a statement.* We want to inspire an open minded reader to think about computing, society, history and philosophy in new ways.

- *Listen and talk to others.* If your background is in X, do not just write for X but look for ways in which X could inform and inspire readers with background in Y.
- *Dare to ask questions and comments as a non-expert.* The book should be accessible, but not oversimplifying. We want to include advanced and novel academic material in a way that welcomes and intrigues a non-expert reader.
- *Dare to take a risk!*

Mini Guidelines for Bertinoro 2019

Here are some mini-guidelines for preparing a first set of chapter sketches to be discussed in Bertinoro. In a plenary meeting, each chapter is presented and discussed in 45 minutes sessions (10 minutes for presentation + 35 for discussion).

- Pick one or more of the four clusters (logic, machines, notations, system) and write a short text (at most 500 words) with your view on the topic -- we are looking for strong focused points on topics that matter to you rather than a broad outline
- Make sure that at least one (skype) meeting has taken place between possible leading authors before Bertinoro.
- Do not prepare drafts with fixed structures, but rather collect a range of ideas that explore a topic from different perspectives. This should encourage contributions from other project members, adding perspectives from other disciplines or rooted in other backgrounds.

For the proposals, we prefer to have:

- a tentative title
- a tentative list of (possible) leading authors (confirmed and non-confirmed)
- a short description of what the chapter is about -- with possible “statement” for the chapter (cfr. appearance and disappearance of the machine)
- an indication of topics and problems to be considered in the chapter with a short explanation for each of them
- open problems/questions for the chapter (e.g. should we really discuss topic x or does it rather belong to another chapter?)
- possible connections to other chapters
- (optionally) ideas for possible programs to start this chapter with (cfr General Guidelines)

Please send us (Liesbeth and Tomas) a document presenting the chapter by **September 15 2019**. This gives enough time to distribute the proposals and plan in more detail the work meetings (and avoid that leading author x is in a parallel session for two different chapters to which (s)he is a contributor).

It is recommended that the wiki is used for developing the chapters in this stage. This allows easy commenting but also (at least in my experience) stimulates free thinking. If you forgot about pwd and/or username, just sent a mail to: liesbeth.de-mol@univ-lille.fr

General Guidelines (to be extended)

1) *Start each chapter with a “program”* It is assumed that each chapter starts out with a specific “program”. This can be a piece of code, a picture of a machine, a patent, a flowdiagram, a citation, a specification, a model program, etc. This program should somehow “capture” the basic idea/statement of the chapter to be (in a way also certain works of art “capture” something).

2) it is hoped that each chapter has both a historical as well as a philosophical, methodological and/or technical dimension. This should not be too strictly interpreted -- it is expected that some chapters will be more historical, some more “philosophical”, others still more methodological, and still others more “technical”.

3) *What is a (computer) program?* It is important not to lose this question out of sight when writing a chapter and so it is very much recommended that, for each chapter, this question is considered towards the end of the chapter (possibly by reconnecting it with the initial program).

4) *Refer to the clusters and/or program modalities* (or, at least, do not forget about them)

5) *Level of difficulty and readership* Try to keep in mind that your reader is not per se someone who has the same background as you have. It is assumed that the reader is someone who is interested enough to make an effort. In other words, when writing a chapter, try to imagine a reader who does not necessarily know much of your specific background but cares a lot for the topic and so, while being willing to engage, say with the methods of philosophy of science, or the methods of software engineering, or formal methods, or, logic, or, etc it should not be assumed that s/he already knows those methods in detail. Some concrete recommendations:

- for certain topics that require some explanations (whether they be technical or historical) that would distract too much from the main line of argument but are nonetheless considered necessary to understand and follow that argument, it is recommended to use an intermezzo (this can be “technical”, historical, methodological, philosophical, etc).
- Allow yourself to question the “idiosyncrasies” of your own background by confronting it with the perspectives and methods from co-authors (of the chapter or the book).
- when reading one of the other chapters (drafts) it is very much recommended to take notes of things you do not understand or which were quite difficult to read. It is quite normal for this project that you do not always understand what is the fuss about in a given contribution -- cfr the right to ask *possibly* naive questions.
- if some passage requires a too long intermezzo to motivate it and make it understandable, than perhaps it is best to drop it and use it for another paper.

Guidelines wrt Cluster chapters

1. A cluster chapter is written by a number of leading authors but it is expected that these are written very collaboratively in a way which welcomes also perspectives from others.
2. A cluster chapter is not a survey nor an introductory chapter. It is general in the sense that it is a basic aspect of the project's methodology.
3. Cluster chapters are supposed to develop a set of basic topics wrt the main question of the project which allows to understand and clarify how the nature of the relation (historical and systematic) of the cluster wrt to (a) computer programs (2) the other clusters.
4. Dare to be self-critical about your cluster by confronting it with the other clusters
5. for the sake of coherence, it is asked that each cluster chapter asks the "what is" question wrt the cluster. In this context, referring to literature that is independent of computer programs but considers this question nonetheless may allow to demarcate more clearly but also to open up the topic (for instance, notion of "systems" in biology; notion of "Machine" in the OED)
6. for the sake of coherence, it is asked that each cluster considers the two main obstacles discussed in the project text (communication gaps and opacity) from the perspective of the cluster. See the project text for more details.