Business Process Modeling and Practical Ambiguity

the gap between organized work processes and computer procedures

Bertinoro, October 2\textsuperscript{nd} 2019
Henri Stephanou
Paris-I University
### Metaphors or conceptual identity?

<table>
<thead>
<tr>
<th>Organized (Human) Work</th>
<th>Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Procedures</td>
<td>1. Procedures</td>
</tr>
<tr>
<td>2. Processes</td>
<td>2. Processes</td>
</tr>
<tr>
<td>3. Instructions</td>
<td>3. Instructions</td>
</tr>
<tr>
<td>4. Rule following</td>
<td>4. Rule following</td>
</tr>
<tr>
<td>5. Implementation</td>
<td>5. Implementation</td>
</tr>
<tr>
<td>7. ...</td>
<td>7. ...</td>
</tr>
</tbody>
</table>
Procedures and Processes

Procedure: a normative description of a process, basically taking the form of instructions directed to an agent, to achieve a purpose (usually known by the agent)

→ A representation, symbolic (text or diagram) – susceptible of computational treatment

...But what about processes? ... ... All computing processes are executions of procedures (sometimes defined very indirectly).

But is this the case with human work processes? Do all human purposeful and rational processes really suppose a procedure, even an implicit one, which the working agent “instructs to himself” ?...
Strong evidence to answer positively

Historical Evidence

Economic Evidence

Ethical claims

Conceptual Analysis (Business Process Modelling)
Deep common roots: Prony & Babbage (I)

« il me vint bientôt à la pensée d'appliquer à la connection de ces Tables la division du travail, dont les Arts de Commerce tirent un parti si avantageux pour réunir à la perfection de main-d'œuvre l'économie de la dépense et du temps » (Prony, quoted by Babbage, 1846)

“It came soon to my mind to apply to the connection of these tables the [principle of] the division of labor, from which the Commercial Arts take such a great advantage, gathering the perfection of work to the economy of cost and time”
Von Neumann and Goldstine recognized that there was nothing in their description of the planning stage that was specific to automatic computing, and the details of their proposals reflected established practice in the organization of largescale [human] computing [...]

This continuity of practice should come as no surprise. All the developers of the automatic machines of the early 1940s had had extensive experience in carrying out or supervising large-scale manual computation. What was new was the need to be completely explicit about the operations that the machine would carry out.”

“The instructions which govern this operation must be given to the device in absolutely exhaustive detail.” (Von Neumann 1945)
A meta-industry

Size of the Global IT industry, 2019: $5T

Size of Pharma: $1.2T

Size of Aerospace: $0.9T
Ethical questions: the techno-pessimist story

“Organization is nothing but technology applied to social, economic or administrative life. [...] It leads to standardization and rationalisation [...] We can be assured that its effects are the assimilation of everything to the machine, that its ideal is the mechanization of everything technology meets ».
Ellul, 1960, p. 9

“Management [...] functions as does any technology. It is not made up of mechanical parts, of course. It is made up of procedures and rules designed to standardize behavior. We may call any such system of procedures and rules a technique; [...] Like so much of our machinery, they become autonomous. There's the rub. In a Technopoly, we tend to believe that only through the autonomy of techniques (and machinery) can we achieve our goals.”
Postman, 1992/1993, p.141
Plan of the presentation

Introduction

Part I: The conceptual evidence from Business Process Modelling

How can human work processes be modelled into computerized procedures?

Part II: The persisting gap between organized work processes and computerized procedures

(even while discounting the « human » and « emotional » parts of human work – focusing on rational behavior)

Conclusion
Part I:  
The conceptual evidence from Business Process Modelling

A few significant Milestones

1. The good old flowchart
2. The advent of ERP reference models
3. The holistic view of UML
4. BPMN and BPEL

Common thread of our investigation:

What does it take to describe organized work processes “in absolutely exhaustive detail”? (Von Neumann)
Part I

1. The good old flowchart
Part I

2. the ERP revolution

Source: Davenport 1998
Illustration: SAP reference map – global view
Part I

2. ERP process reference models

Illustration: SAP reference map – module view
1. A customer asks for a loan, providing name and amount info
2. Two services are involved:
   a) A risk assessor which can approve the loan if the risk is low
   b) A loan approver which checks the name and approves/disapproves the loan
3. The reply is returned to the customer
Part I.4: BPMN and BPEL

**BPEL**: basic and structured « activities » (commands)

- **receive**: Do a blocking wait for a matching message to arrive
- **throw**: Generate a fault from inside the business process
- **reply**: Send a message in reply to a formerly received message
- **rethrow**: Forward a fault from inside a fault handler
- **invoke**: Invoke a one-way or request-response operation
- **exit**: Immediately terminate execution of a business process instance
- **assign**: Update the values of variables or partner links with new data
- **wait**: Wait for a given time period or until a certain time has passed
- **validate**: Validate XML data stored in variables
- **empty**: No-op instruction for a business process
- **flow**: Contained activities are executed in parallel, partially ordered through control links
- **pick**: Block and wait for a suitable message to arrive (or time out)
- **sequence**: Contained activities are performed sequentially in lexical order
- **forEach**: Contained activity is performed sequentially or in parallel, controlled by a specified counter variable
- **while**: Contained activity is repeated while a predicate holds
- **if then else**: Select exactly one branch of activity from a set of choices
- **repeatUntil**: Contained activity is repeated until a predicate holds
- **scope**: Associate contained activity with its own local variables, fault handlers, compensation handler, and event handlers
Part I.4: BPMN and BPEL

BPEL scopes
Part I: Summary

What does it take to model human work processes

Human work processes are much more complex than just sequences of activities

To define them « in absolutely exhaustive detail » requires
  • Advanced syntax and semantics allowing to compose with actions, events, sequences, forks, parallel flows – structured operational semantics needed
  • A common data referential OR rules to exchange data between each one’s referential
  • The formalization of their « collaboration » with each other: with whom to collaborate, what messages can be exchanged, to which events to respond

In summary, a process can only become a formal object when its full context (or « scope ») has been exhaustively defined
  • BPEL achieves this aim by requiring each process to fully define its own scope: XML logic of self-declaring objects
Part II: The persisting gap with human work processes

How humans see their processes...
Part II

Two examples of practical ambiguity

1. **Leaflet customization at packaging**
   - Packaging is the end of the production process, where we put the product in the box with the invoice.
   - Many subsidiaries also inserted a leaflet, or two leaflets inside the box. This a process variant; it is unspecified. It is not said whether it is allowed or not.
   - This was no issue, until marketing asked to customize the leaflet insertion depending on the customer profile, and the plant manager then asked IT to prompt the worker to the right leaflet.
   - The issue is to make explicit an action item which does not exist at all in the process referential!

2. **Saturday production issue**
   - Customers can cancel their purchase up to one day after their order, because you can expect safely that the production will not have been launched until then.
   - However, in this case, orders passed the Friday evening could be cancelled only on Monday (workday for customer service) whereas production worked on Saturdays. So cancellations were arriving too late.
   - What production was doing was to postpone as much as possible the production of Friday lenses until next Monday, but the new automated system did not have this rule.
Part II
The value of ambiguity in organizations

In many cases managers prefer not to standardize the behaviors of their teams, allowing them:
- to customize or personalize the products or services to customers’ needs;
- to adapt to local or temporal circumstances, to stay agile;
- to develop self-improvement initiatives;
- to postpone decision-making until necessary – (cf. « laziness » in programming)

“Researchers […] have embraced the traditional philosophical and scientific view that purposeful action is planned action. Yet even casual observation of purposeful action indicates that, as common-sense formulations of intent, plans are inherently vague. To the Cognitive Scientist, this vagueness is a fault to be remedied […]. For situated action, however, the vagueness of plans is not a fault but, to the contrary, is ideally suited to the fact that the detail of intent and action must be contingent on the circumstantial and interactional particulars of actual situations.” (Suchman 1987, p. 123)

« A [cooperative work] protocol is deliberately under-specified with respect to […] factors that can more efficiently and effectively be left unspecified, typically until a later stage. The protocol must, to use the apt phrase of Bowker and Star, be defined at ‘an appropriate level of ambiguity’” (Schmidt, 1996/2007, p. 112)

More broadly, ambiguity has been recognized as a behavioral pattern of organizations in the management literature
- According to (Weick, 1979) firms must continuously strike a balance between stability (embodied by procedures, rules, formal hierarchies) and adaptation (enabled by ambiguity, fluid relationships between actors, informality in decision-making)
Part II

Ambiguity can always be resolved... at a cost

If our infringing the rule had reasons, then we can model that into a new rule

Let us take context $C$ composed of $(c_1, c_2, \ldots, c_n)$ variables (events, messages, partners...) and a process $P$ composed of $(p_1, p_2, \ldots, p_m)$ steps.

I will write:

$$C : P$$

If ambiguity is at step $p_k$, then we could put a short-circuit $SC$ there, allowing humans to take control whenever fit. For example they can put aside Friday orders in the production schedule. So we have:

$$C : P (p_1 p_2 \ldots p_{k-1} - SC - p_k \ldots p_m)$$

But the ideal thing is to make it automatic. How can I do that? I add the relevant variable $c_{n+1}$ (for example weekday) in the context, and then I can customize the step $p_k$ into a variants $p_k'$:

$$C' (c_1 c_2 \ldots c_n c_{n+1}) : P' (p_1 p_2 \ldots p_{k-1} (c_{n+1} ? p_k / p_k') p_{k+1} \ldots p_m)$$

The conclusion is that any rational behavior can be captured through a process model.

However, the complexity can increase rapidly – as contexts become fatter and new rules can be inconsistent together and new rules added to set priorities. It is called the “process variability issue” (Rosa et al., 2017)
Part II
Programming as environment definition

So the key point is to decide how granular you want your contexts to be in the model, and how specific you want your actions to be.

• You may want to rule out any ambiguity and model all possible behavior, but in unstables contexts this may not be the most judicious thing to do

Process modeling appears therefore as matching a context (signals to the process – including goals) with an alphabet of actions (basic capabilities of the process)

This is exactly how Herbert Simon defines an artifact, as an interface between an external and internal environments (which are defined by it):

« An artifact can be thought of as a meeting point – an ‘interface’ in today’s terms – between an ‘inner’ environment, the substance and organization of the artifact itself, and an ‘outer’ environment, the surroundings in which it operates. If the inner environment is appropriate to the outer environment, or vice versa, the artifact will serve its intended purpose. » (Simon 1969/2008, p. 6)

The essential part of programming – as well as managing – is to specify these two environments – define the basic language – signals and actions – that the program will operate on
Rational work processes can always be modelled into computer processes: conceptually identical concepts

This modelling however captures « reasonable » behaviour only under a certain modelling of the context

If contexts are unstable or diverse, humans will adapt rationally – make up new rules on the go which will make the procedure outdated, which is what we call practical ambiguity

Human reasonable behaviour then always exceeds and precedes procedures, although it can always be described through them ex-post

*Computing processes always execute a procedure; Human processes make up procedures on the go*

Reason is therefore not a blind, one-sided force for proceduralization ; in many cases depriving humans from decision-making is not the reasonable – nor profitable – thing to do
could new developments in technology make computers compete with humans in ambiguity?

1) The value of ambiguity may diminish as systems interconnect together and humans driven out of the value chains
   • In the case of the Friday production issue, the solution was to let customers post cancellations on the website and connect the website directly to the plant (which was not easy)
   • self-driving cars could be drastically simplified if they were only allowed on the streets and could communicate between them and with road signage)
   • Cf. ethical worries about the systemic nature of technical progress

2) Could artificial intelligence learn to behave ambiguously?...