Production of social statistics…goes social![♣]

Jacopo Grazzini and Pierre Lamarche[♦]
({*jacopo.grazzini,pierre.lamarche*}*@ec.europa.eu*)

Keywords: *statistical production system, data workflow, hybrid design, open algorithms, reusability and reproducibility, control and maintenance, sharing and collaboration, GSBPM, CSPA, software-and technology-agnostic implementation, micro SOA.*

1. OBJECTIVES

The scope of this paper is to present a practical framework adopted for the integration of software applications into a statistical production chain. The focus is the actual implementation of a high-level collaborative platform aiming not only at producing social statistics, but also at further fostering experimentation and analysis in that field. In doing so, we strongly support the (obvious) claim of **[1]** that *"the modernisation and industrialisation of official statistical production needs a unified combination of statistics and computer science in its very principles"*.

Motivated by the consensus that processes – in particular statistical processes **[2]** – for data-driven policy should be transparent **[3]**, we naturally promote open, reproducible, reusable, verifiable, and collaborative software development and deployment in a statistical organisation. Beyond just devising guidelines and best practices, we show how the platform is implemented for the production of social statistics. For that purpose, we adopt, as recommended in **[4]**, a reasonable mix of *bottom-up* (from low-level scope to high-level vision) and *top-down* (from black-box process models to traceable functional modules) designs, so as to *"think global, [and] act local"* **[5]**. In building the parts while planning the whole, we provide with a flexible and agile approach to immediate needs and current legacy issues, as well as long-term problems and potential future requirements for statistical production **[6]**.

2. CONTEXT

The development and deployment of a production system is not a trivial task. It can be successful only if data is managed properly through its life-cycle, by setting the right context to perform well-designed operations integrated into the right processes – *e.g.*, from collection to dissemination through validation, integration, exploration and analysis – all in a timely fashion **[4]**. Several challenging issues which accompany the practical implementation require specific focus *e.g.* in modelling and programming, in handling and managing data, in running the execution of processes, to mention a few. A production system needs to reach certain standards **[7]** in terms of: *interoperability*, particularly when considering the mix of structured and unstructured data, qualitative and quantitative data; *accessibility*, since continuous operations may need to be achieved with no human intervention; *effectiveness and robustness*, to enable routine computational tasks to run automatically, effectively and consistently on data; *scalability and timeliness*, so as to provide quality statistics quickly and efficiently; *flexibility and extensibility*, because implementation usually happens without basic knowledge of the runtime targets and since it is often desirable to satisfy a wide range of applications as well as evolving requirements. On that latter aspect, the long-term maintenance and

[♣] …and beyond: supporting material for this article is available here.

[♦] European Commission – DG ESTAT.

extension of a production system may reveal burdensome due to legacy issues, *e.g.* software components with customised design have changed or have become obsolete **[8]**.

Apart from other technical and methodological factors, a statistical production chain is a complex system **[1]** . It is usually difficult to describe production tasks in a homogeneous fashion since there is often a large number of intricate components and ambiguous operations. As a result, a human factor often adds to the complexity since inaccurate and out-of-date information may hinder efficient coordination and communication between the actors of the production chain – designers, implementers, and maintainers, not to mention analysts. Under that aspect, a more integrated approach to statistics production is, again, desirable so as to optimise the data production workflow.

3. ADOPTED DESIGN PRINCIPLES

Some practical requirements are regarded as sound principles to gear the development of IT systems (*e.g.,* used in the production of social indicators) in a public statistical organisation in order to ensure complete transparency.

- *Openness* is one crucial aspect, so as to favour reusability and reproducibility **[9]** . It shall however not only be interpreted as in *"open source software"* only, but rather as in *"open source code"* and *"open algorithm"*. Open source software solutions are obviously preferred **[10]** – since it guarantees openness and adaptability, though still susceptible to downsides – however the actual implementation is often tied to legacy proprietary software in prominent and wide use in statistical offices **[11]** .

- *Reproducibility* **[12]** is an attainable minimum standard to ensure that the processes and computational workflows can be replicated. It contributes to consistency checks and quality assurance of statistical products. Actually, we aim at complying with the rules: *"for every result, keep track of how it was produced"*, and: *"provide public access to scripts, runs, and results"* enounced in **[13]**.

- Another important, already mentioned, requirement for integrated components is their *reusability, i.e.,* they should be applicable in contexts/domains other than the ones for which they have been developed **[12]**. Again, a critical barrier is, in many cases, that the source code used for a given application is not available – or written in a way that defies its reusability.

- *Verifiability* is also desirable **[9]** : given any model and its input configuration, it should be possible not only to regenerate the output, but also to fully test and validate the underlying methodology. This will naturally derive from reproducibility.

- Full *in-house control* is a basic requirement that favours the continuity of operations, but also supports their future evolution **[7]**. In principle, no link involved in the production chain should turn into a bottleneck, *e.g.,* because some expertise is drained off and/or because the software system has not kept track of producers' actions in concrete form. Additionally, a strong control of the practical and effective implementation of statistical methods and techniques is also required. Statistical software packages – may they be open or commercial-off-the-shelf – generally implement a broad range of techniques but do so in an *ad-hoc* manner, leaving users at a disadvantage since they may not understand all the implications of a given process, or how to test the validity of results produced by the software. Another key issue to consider is therefore the comprehensiveness of the statistical information that should be achieved through clear and efficient implementation.

- Finally, *sharing* needs to be supported **[9]** : computational resources facilitate in-house participation and contribution. Any skilled person could modify the code to suit his/her needs, learn from its use and further contribute to its improvement. It can help adopting new efficient ways of working in production – avoiding developments to start from scratch every time – as well as collaborating beyond the statistical organisations **[12]**. Indeed, outside communities can also provide further robustness and additional features to the implementation.

Overall, a greater transparency in designing production processes is expected to result in a better grip on the quality of the output statistical products **[4][7]**.

4. PRACTICAL IMPLEMENTATION FRAMEWORK

As underlined in **[4]**, adopting a more integral approach for the design of statistical production processes creates, from the corporate point of view, opportunities for economies of scale in process design, statistical methods implementation, IT components development, and so on. From a top-down perspective, it is particularly useful to raise the level of abstraction when designing the production workflow, so as to consider generic processes instead of looking at single tasks **[7][14]** . In that aspect, the whole data production chain can be organised into processes – and sub-processes – following the classification framework provided by the Generic Statistical Business Process Model (GSBPM, see **Figure 1**). Raising the level of abstraction – without necessarily increasing the level of standardisation – is also essential for enabling flexibility and extensibility, considering that a production system will most likely change during its lifetime **[5]**. For that purpose, a *"plug and play"* design **[4]**, based on *modular and customisable* components which are easy to assemble into a reliable processing system, can be adopted. First, this design makes possible both vertical and horizontal integration by possibly adding up already existing and/or new processes and operations corresponding to different domains of the GSBPM **[6]**. In addition, the use and management of metadata to describe not only data, but also programs and tools, can facilitate the reuse of these modular components, or parts of them. Still, when designing such a modular production process, special attention needs to be paid to the specification of the statistical methods and the deployment of the IT modules in which the statistical methods are encapsulated **[4][8]**. Actually, it shall be considered from both the business and the service perspectives since there is no one-to-one matching between statistical processes and IT modules. In doing so, careful analysis of the scope and boundaries of each individual module – may it be IT or statistical – and the level of *granularity* of the set of modules, as well as their interdependencies, should also be performed **[7]**. Second, by focussing on the overall workflow instead of single tasks, the modular design further releases the constraint on the choice of a programming language: within a given module, it should be possible for the producer to implement a given statistical operation in whatever programming language that fits best. Obviously, this *"agnostic"* approach will have a cost since it introduces further constraint on the implementation: it becomes necessary to ensure that the statistical products obtained using different software/technologies are identical, *i.e.* that processes are consistent and robust; it is also essential to guarantee that different modules – ideally provided in a service-oriented manner **[14] [15] –** fit seamlessly together and can communicate between each other. Nevertheless, it leverages the opportunities that new technologies offer for working more effectively and it supports software (and hardware) independence in view of future migration, thereby alleviating the potential burden of legacy. It also offers additional control since it guarantees not only the validity of the statistical products, but the methodological choices made for implementation as well – *e.g.*, on the way a Gini index is calculated, a quantile is estimated, sampling is performed…

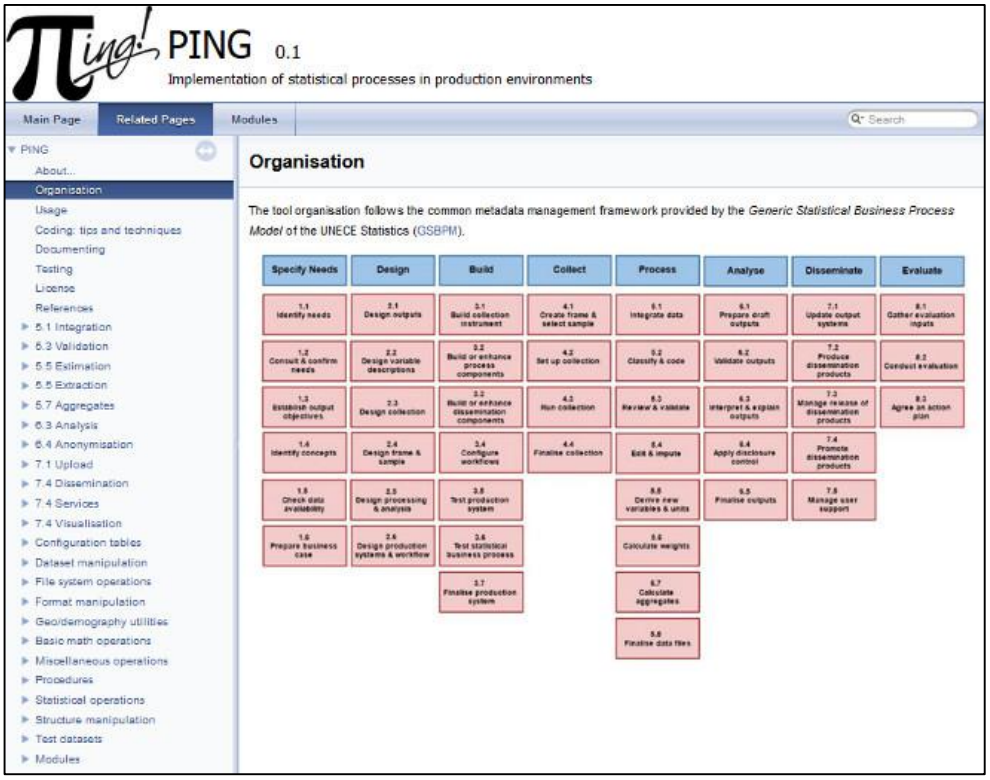

Figure 1 - A snapshot of the user-friendly documentation of PING platform (available here). The organisation matches that of GSBPM. In addition, when applicable, processes, sub-processes and operations are described through BPM graphical representation.

When applying an integral approach to statistics production, neither does everything need to be designed at the same time, nor does this design need to be absolutely right prior to the actual implementation. We further combine the high-level process-centric approach above with bottom-up considerations, *e.g.*, specific methodological and technological aspects of the processes are also taken into account through continuous implementation **[1]** . In practice, it means that operations along the production chain are modularised not only to support the whole production cycle, but also to solve immediate ad-hoc issues. This leads notably to a preference for simpler and smaller process modules. Small granularity makes it easier to change, add or remove features and capabilities to a module at any time, and to build complex operations from simple parts **[4][8][14]** . It increases the flexibility of process modules, as well as their performance, which can be controlled through the introduction of various configuration and parameter settings. The degree to which a module is then configured and parameterised is mostly a methodological issue and may imply some judgmental choices. There is an obvious trade-off between the scope of application (flexibility), the ease of application (reusability), the efficiency, and the simplicity of the modular components **[7]**. However, a limitation on the range of statistical methods configurable and available for each process can be imposed (see example of **Figure 2**). This way, by starting small and building quickly, we develop a library of validated statistical methods and IT components, and gain experience from its deployment. Beyond being exemplified, methods and programs are fully tested to ensure their reliability **[16]**. Because maintainability and consistency may suffer from the increase of flexibility, they also need to be consistently documented: documentation of both methodological concepts and computational aspects not only enhances reproducibility but also creates the basis for quality assurance **[12]**. In addition, the use of versioning system helps at differentiating between configurations used in production and

those under development. Overall, our goal is not only to tackle issues dealing with the reuse of IT solutions from a corporate point of view, but we also aim at sharing and reusing code by making guidelines for developing, documenting and testing programs available. This way, users – in statistical offices, but not only – shall become producers, say *produsers*, and potentially contribute to new experiments and help develop more advanced analysis methods. This collaborative approach reduces both the risk of having no-one available to maintain or update the system, and the cost of the necessary testing and audit procedures needed for high-reliability software. Altogether, it contributes to a more holistic and extensive approach to production systems in statistical offices.

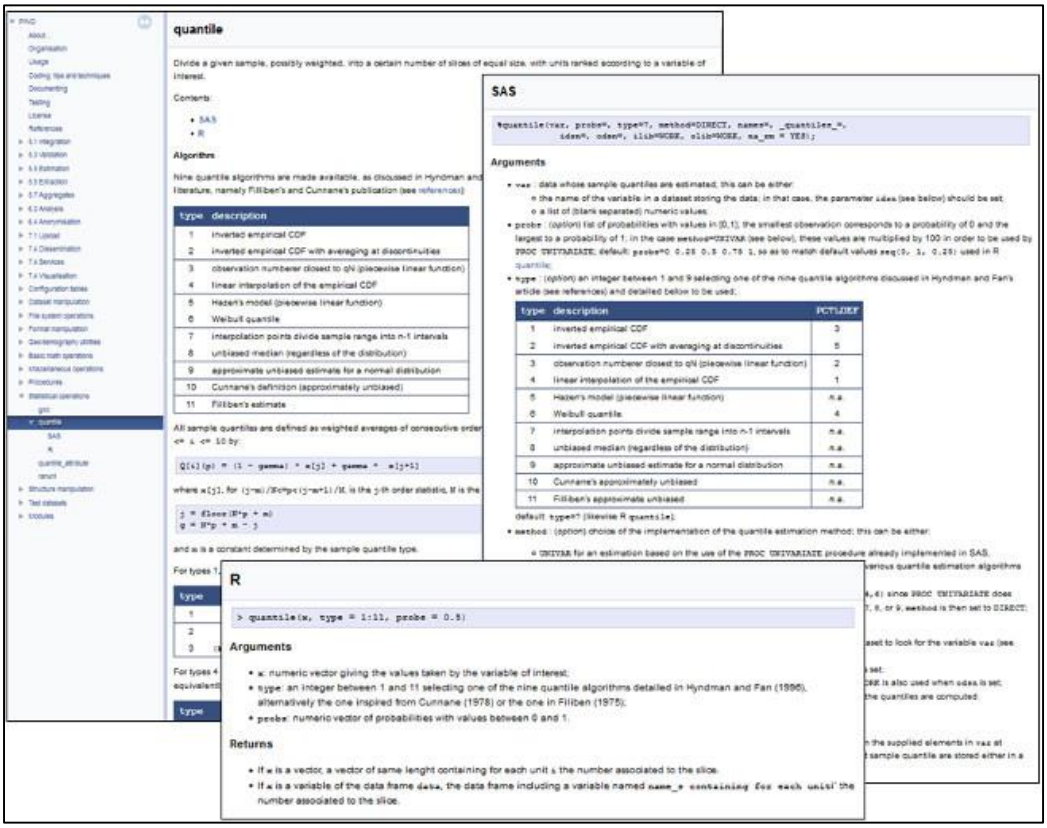

Figure 2- A simple statistical operation (quantile estimation) is consistently implemented (i.e., following well-defined algorithms) in different programming languages and made available to produsers with a limited range of parameters. Currently, both R and SAS developments are integrated onto PING platform. In the future, unnecessary duplication shall be avoided, and, instead, a software-agnostic service could be provided (*e.g.*, through a micro SOA [15]).

5. CONCLUSION: CURRENT STATUS AND FUTURE PLANS

This paper presents a reference framework – designed as a mix of bottom-up and top-down approaches – which gives overall guidance for the integration of software applications into a statistical production chain, as well as a number of principles that underpin this framework. Currently, specific statistical production processes are still being incorporated within this framework, while shared IT modules are being developed. The former becomes more attractive the more standard IT modules are available to support it, and the benefit of the latter increases with each process that is designed to use it. In the future, we plan to explore ways of moving along the dimension of scalability while again maintaining the generality of the platform. We believe that progress will be greatly facilitated by sustained and strong interaction between all *produsers*, with expertise cutting across multiple domains. Further, interrelation between standards for business process model and information model will also be explored.

REFERENCES

[1] Salgado D. (2016): A modern vision of official statistical production (online access).

[2] Sutherland W.J. *et al.* (2013): Policy: Twenty tips for interpreting scientific claims, *Nature*, 503:335-337, doi:10.1038/503335a.

[3] Bertot J.C. *et al.* (2010): Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies, *Government Information Quarterly*, 27(3):264-271, doi:10.1016/j.giq.2010.03.001.

[4] Struijs P. *et al.* (2013): Redesign of Statistics Production within an Architectural Framework: The Dutch Experience, *Journal of Official Statistics*, 29(1):49–71, doi:10.2478/jos-2013-0004.

[5] Lalor T. and Vale S. (2013): Modernising the production of official statistics, *Revista Internacional de Estadistica y Geografia*, 4(2):72-79 (online access).

[6] UN Economic Commission for Europe: Common Statistical Production Architecture, *version 1.5* (online access), Generic Statistical Business Process Model, *version 5.0* (online access), Generic Statistical Information Model, *version 1.1* (online access).

[7] Poirier C. *et al.* (2013): Discussion, *Journal of Official Statistics*, 29(1):193–201, doi:10.2478/jos-2013-0014.

[8] Seyb A. *et al.* (2013): Innovative Production Systems at Statistics New Zealand: Overcoming the Design and Build Bottleneck, *Journal of Official Statistics*, 29(1):73–97, doi:10.2478/jos-2013-0005.

[9] Ince D.C. *et al.* (2012): The case for open computer programs, *Nature*, 482:485-488, doi:10.1038/nature10836.

[10] Templ M. and Todorov V. (2016): The software environment R for official statistics and survey methodology, *Austrian Journal of Statistics*, 45:97–124, doi:10.17713/ajs.v45i1.100.

[11] Billings T.E. and Bankston E. (2013): Bridging the gap between SAS applications developed by business units and conventional IT production, *Proc. SUGI* (online access).

[12] Kreuter F. (2013): Discussion, *Journal of Official Statistics*, 29(1):165–169, doi:10.2478/jos-2013-0010.

[13] Fiorenza P. *et al.* (2013): Ten steps to leveraging analytics in the public sector, in *Unlocking the Power of Government Analytics*, pp.29-32 (online access).

[14] Bruno M. *et al.* (2016): CORE: a concrete implementation of the CSPA architecture, *Statistical Journal of the IAOS*, pp.1-6, doi:10.3233/SJI-160977.

[15] Dragoni N. *et al.* (2016): Microservices: yesterday, today, and tomorrow, arXiv:1606.04036.

[16] Wilkinson L. (1994): Practical guidelines for testing statistical software, in *Computational Statistics*, Physica-Verlag (online access).