The Primacy of Process Architecture

Stewart Green[1] and Martyn Ould[2]

[1] University of the West of England, Bristol
stewart.green@uwe.ac.uk
[2] Venice Consulting Ltd., Bath
mao@veniceconsulting.co.uk

Abstract. Piecemeal development and support for organisational processes may lead to problems: first, it is difficult to know which processes should be supported, and, second, it is unlikely that piecemeal processes will work well together. The solution seems to lie in first developing a coherent process architecture. However, this raises a further problem: a number of different types of process architecture have been proposed, but it is not clear which one should be used in a given situation. To help to address this problem, the paper describes and evaluates one particular process architecture type and its associated development method, *Riva* [3]. The method was used to create a process architecture for part of a University faculty. It was found to be easy to apply, and, in addition, the resulting process architecture was found to be easy to understand and to use. This result should help others to decide which type of architecture to use when this kind of domain context is encountered.

1 Introduction

Currently, within organisations that are moving towards a process-centric culture, the prevailing tendency is to develop support for organisational processes in a piecemeal manner. But this often leads to a number of problems. First, organisational processes that have been identified piecemeal for improvement and support are unlikely to form the coherent set that is needed to address the organisation's strategy and meet its objectives. This is because, for example, key processes may have been left out of the set. Second, it is unlikely that processes selected piecemeal will work well together.

To address these problems, Harmon, among others, has emphasised the importance of developing and maintaining a process architecture prior to developing computer-based support for particular organisation processes [1]. But once the value of process architectures has been recognised, the next questions are: what comprises an appropriate process architecture, and how are process architectures created? These are important questions to answer correctly because, as Ould [2] has pointed out: "an inappropriate division of organizational activity into processes can easily lead to complex design or models", and "a bad initial division can at best obscure the possibility of radical change to a process and at worst lead to the local optimisation of part-processes, but overall pessimisation

of the total process". A number of different kinds of process architecture have been proposed, each with its own distinctive rationale. For example, Ould has proposed deriving process architectures from the essential entities of a business [2, 3]; Lunn's process architecture is based upon business processes, where a business process is a "logical grouping of events that can be agreed as a fundamental element of a business" [4]; Loucopoulos's process architecture is organised around the goals of an organisation [5]; and Snowden and Kawalek have proposed an architecture [6] based upon Beer's Viable System Model [7].

Currently, it is not clear whether one type of process architecture will come to be viewed as superior to the others for most or all situations, or whether a range of process architecture types is needed. Ideally, a comparative analysis of the different kinds of process architectures is needed in order to resolve this uncertainty. However, in the absence of such an analysis, this paper describes how in the faculty of Computing, Engineering, and Mathematical Sciences (CEMS) at the University of the West of England (UWE), in Bristol, a process architecture has been developed using Ould's *Riva* [3] method. Through this work it is hoped that a clearer idea will emerge of the nature and value both of this particular kind of process architecture, including its strengths and weaknesses, and of its development method. To the extent that this occurs, it can inform the debate about evaluating different kinds of process architecture. In addition, it is hoped that this application of *Riva* may surface any problems inherent in *Riva*, and thus allow them to be addressed.

The paper briefly describes Ould's *Riva* method for creating a process architecture. It continues by recounting how *Riva* was used to create a process architecture for the programme administration team within the faculty. Next the experience of using *Riva* is evaluated, and the main problem encountered is described along with suggestions for its resolution. The paper ends by outlining present and future plans for supporting both the process architecture and the processes based upon it.

2 *Riva* and Its Application

The fundamental rationale for the particular kind of process architecture suggested by Ould [2, 3] is that it is based upon the key entities associated with an organisation. Because a part of an organisation, a University faculty programme administration team, for example, is characterised by the management of awards, students, and modules (among other things), then the process architecture for this organisation should be based directly on these key entities. Furthermore, it is expected that exactly this process architecture will be found wherever these particular key entities are the ones being dealt with by an organisation. There may be differences from organisation to organisation over the way processes comprising the architecture have developed or been designed, but the process architecture itself will be the same. In other words, each set of key organisational entities determines a unique, invariant process architecture.

Ould has created a process, *Riva*, for deriving a process architecture for an organisation [3]. Broadly, *Riva* has the following key stages:

1. Brainstorm the subject matter (essential business entities (EBEs)) of the organisation, or of a part of the organisation.
2. Identify those EBEs (Units of Work (UOW)) that have a lifetime which the organisation must handle.
3. Create a UOW diagram that shows the (dynamic) relationships between UOWs that pertain when one UOW *generates* (or *calls for* or *demands* or *activates* or *requires*) another.
4. For each UOW, hypothesise that there will be in the process architecture a case process that deals with a single instance of the UOW, a case management process that deals with the flow of instances, and a case strategy process that determines the future strategy for both the case and case management processes.
5. Transform the UOW diagram into a corresponding process architecture by turning the relationships between UOWs into relationships between the corresponding case processes and case management processes.

During a one-day workshop held in the CEMS faculty at UWE, *Riva* was used to derive the process architecture for the programmes administration team. During stage one, around one hundred EBEs were identified through brainstorming. They included, for example, *student, field, school, visiting lecturer* and so on. During stage two, this list was reduced to about thirty UOWs, including, for example, *award definition, student, programme plan, module run,* and so on. During stage three, a number of dynamic relationships were identified. For example, the *programme plan* was seen to generate *module runs*; and each of these, in turn, was seen to generate both *submissions* and *assignment and exam paper definitions*. Some UOWs were also seen to be stand-alone: they did not participate in any dynamic relationships. These included, for example, *award handbook* and *induction week*. Figure 1 shows a part of the UOW digram that was produced during this stage.

Stage four was enacted by, for example, hypothesising that there is a case process for handling an instance of a module definition, and an associated case management process for handling the flow of instances of module definitions. Similar pairs of processes were hypothesised for each UOW. (Case strategy processes were not deemed to be relevant to this part of the organisation.) In stage five, *Riva's* transformation rules were used to enable a process architecture to be built from the UOW diagram. Figure 2 shows a part of the process architecture that was actually built.

Figure 2 shows, for instance, that during the life of one instance of the *Handle a module run* case process, one or more requests to handle a submission may be made to the running instance of the case management process *Manage the flow of all submissions*. For each request, the case management process decides when to start a new instance of the case process *Handle a submission*. As each

[3] *Riva* also includes an approach for modelling processes [10]

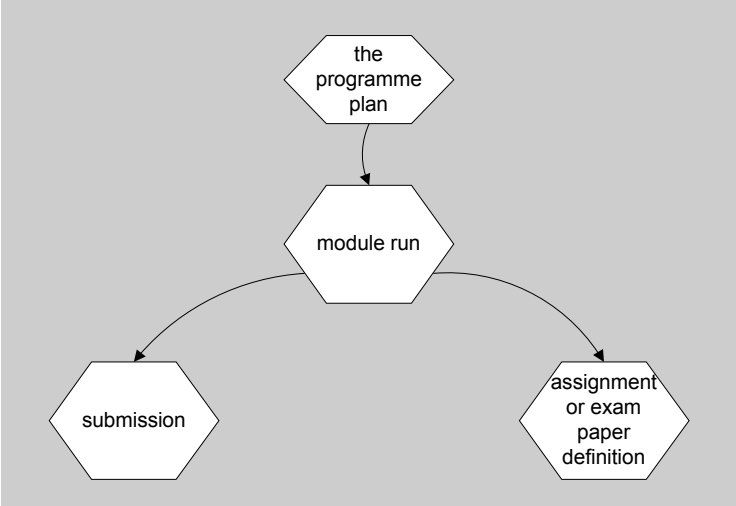

Fig. 1. A part of the UOW diagram

instance of the latter completes, it delivers a set of marks back to the *Handle a module run* process. Figure 2 also illustrates that not all UOWs have a case management process: there is no case management process for *Handle a module run*, for instance; this is because module runs come one at a time.

3 Evaluating *Riva*

On the positive side, first, *Riva* was found by the workshop participants to be relatively straightforward to enact. In general, it is expected that the ability of a workshop facilitator, and the general ability and preparedness of workshop participants, will be the main factors in determining the ease of use of *Riva*. Second, the process architecture produced by *Riva* seemed to be understood by the workshop stakeholders, who later successfully engaged with it both by matching pre-*Riva* processes to the architecture, and by customising the names of process architecture processes. However, it remains to be seen to what extent the rest of the faculty will understand the architecture.

So *Riva* was relatively easy to use, and the resulting process architecture was understandable. However, one important question has been asked about the method: in step one, how is it known that all the EBEs have been identified? As the EBEs are the drivers of *Riva* and, therefore, of the process architectures it produces, it is important that all the EBEs for an organisation are identified. To facilitate this there is a need both for heuristics to find EBEs quickly, and for more explicit rules for determining EBEs from non-EBEs.

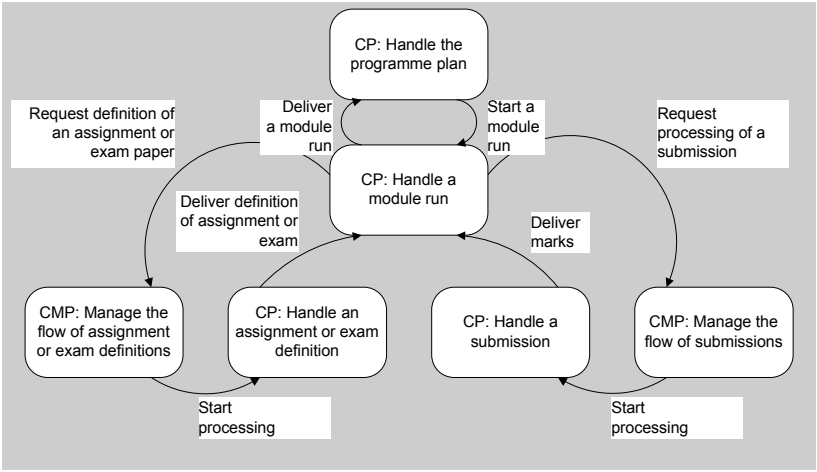

Fig. 2. A part of the process architecture

4 Using the Process Architecture Generated by *Riva*

Currently, the UWE programmes administration team process architecture is being used to identify which processes to describe, model, and publish on the faculty's intranet for the use of staff and students. Process models like this are passive models as defined by Warboys et al. [8]. In the near future, it is intended that the process architecture and set of process models will be used to create an IS strategy [3, 9, 11] for the programme administration team, and, from thence, to create the identified IS applications to support their work. In the longer term, it is intended to migrate to a system that will allow process models to be run as programs by a process enactment system. Process models like these will be active models as defined by Warboys et al. [8].

It is also intended that the process architecture will be kept current. From the rationale for this particular kind of process architecture, it may be deduced that the architecture should be reviewed whenever the programme administration team is asked either to handle a new UOW, or to stop handling an existing UOW. In addition, processes and their supporting IS systems, both derived from the process architecture, will also be reviewed regularly with a view to improving them. For example, if a new enabling technology becomes available, processes will be reviewed to see whether and how they can be modified to exploit it.

5 Conclusion

The paper has described the experience of using one kind of development method, *Riva*. It was successfully used at UWE to create a process architecture for the programme administration part of the CEMS faculty. The architecture produced

is well understood by those associated with producing it, and has been used as the starting point for modelling key faculty processes. However, some challenges for *Riva* remain: in particular, it is recommended that the method be strengthened by the inclusion of heuristics and rules for facilitating the identification of all the EBEs for an organisation.

It would be useful to extend this work by providing a critical comparative analysis of different kinds of process architecture and their associated development methods. Such work would involve the identification of a comprehensive set of process architecture types, and the evaluation of each type according to a set of criteria. The criteria would include the following, for example: the soundness of a process architecture's theoretical underpinnings, including its axiomatic assumptions; the ease of use of a process architecture's derivation method and its associated notation; the completeness of a process architecture's verification method; the utility of a process architecture for deriving processes and acting as a medium of communication—including to non-professionals; the suitability of a process architecture for small organisations, medium organisations, and large organisations; and the ease of maintenance of a process architecture.

References

1. Harmon, P.: Business Process Change, Morgan Kaufmann, (2003)
2. Ould, M.: Designing a Re-engineering Proof Process Architecture, Business Process Management Journal, **3(3)**, (1997), 232–247
3. Ould, M.: Basing an Information Systems Strategy on the Organization's Process Architecture. In: Grover, V. (ed.): Process Think: Winning Perspectives for Business Change in the Information Age, Idea Group Publishing, (2000), 209–226
4. Lunn, K., Sixsmith, A., Lindsay, A., Vaarama, M.: Traceability in Requirements through Process Modelling, Applied to Social Care Applications, Information and Software Technology: Special Edition on Business Process Modelling, **45(15)**, (2003)
5. Kavakli, V., Loucopoulos, P.: Goal-driven Business Process Analysis Application in Electricity Deregulation, Information Systems, **24(3)**, (1999), 187–207
6. Snowden, R., Kawalek, P.: Active Meta-Process Models: a Conceptual Exposition, Information and Software Technology: Special Edition on Business Process Modelling, **45(15)**, (2003)
7. Beer, S.: The Brain of the Firm, Wiley, (1981)
8. Warboys, B., Kawalek, P., Robertson, I., Greenwood, M.: Business Information Systems: A Process Approach, McGraw Hill, (1999)
9. Odeh, M., Kamm, R.: Bridging the Gap Between Business Models and System Models, Information and Software Technology: Special Edition on Business Process Modelling, **45(15)**, (2003)
10. Ould, M.: Business Processes: Modelling and Analysis for Re-engineering and Improvement, John Wiley, (1995)
11. Holtzblatt, K., Beyer, H.: Contextual Design: Defining Customer-Centered Systems, Morgan Kaufmann, (1998)