Integrating Cross-Organisational Business Processes Based on a Combined S-BPM/DSM Approach

Udo Kannengiesser
Metasonic GmbH
Münchner Str. 29 – Hettenshausen
85276 Pfaffenhofen, Germany
udo.kannengiesser@metasonic.de

***Abstract*—This paper addresses the issue of cross-organisational process integration using the Design Structure Matrix (DSM) approach from engineering design. This approach includes a set of generic techniques for minimising iterations within processes, thus reducing the impact of rework on processing times both within single processes and across interconnected processes. The paper focuses on the latter: How can the DSM be used for aligning processes that are interconnected yet performed by separate organisations? The paper shows that the subject-oriented approach to Business Process Management (S-BPM) serves as an enabler of DSM-based process integration. An example of using the combined S-BPM/DSM approach for cross-organisational process integration is presented to demonstrate its applicability and benefits.**

Keywords—Business Process Integration; Business Process Improvement; Business Process Modelling; Subject-oriented Business Process Management (S-BPM); Design Structure Matrix (DSM)

I. INTRODUCTION

In today's networked economy, cross-organisational and cross-company processes play an increasingly important role for value creation [1]. These processes are composed of a multitude of services provided by various business partners, suppliers, competitors, public entities and other organisations. To ensure smooth interoperation in such highly distributed value networks, the issue of process integration arises: How can the individual pieces of a process be composed across organisational boundaries to minimise communication and coordination effort [2]?

A key issue for process integration is the minimization of iterations, as they represent rework that cause delays and cost overruns. This paper addresses this issue using an approach from engineering design: the design structure matrix (DSM) [3, 4, 5]. The DSM has been used for decomposing processes, analysing the relationships between the components, and recomposing them to minimize iterations. While this approach has been developed and applied for managing processes in engineering design and product development, it is a generic tool that can potentially be used in any process domain characterised by complex interactions between process entities. Yet, to date this tool has remained almost unnoticed by the business process management (BPM) community, most likely due to the conflicting modelling paradigms of DSM and the mainstream BPM approaches: The DSM models processes as flows of information, whereas most BPM methods describe processes as flows of control. To make the DSM accessible to BPM practitioners, this paper uses the Subject-oriented BPM (S-BPM) approach [6] whose emphasis on representing the communication between process participants is consistent with the information-flow paradigm of DSM. A combined use of S-BPM and DSM is proposed to fully include cross-organisational process integration in the (S-BPM) process lifecycle, enabling the validation, implementation and execution of integrated processes without manual model transformations.

The paper is organised as follows: Section 2 introduces the DSM and the basic techniques it provides for process improvement and process integration. Section 3 outlines the S-BPM approach and its ability to model communication relationships between process elements, as needed for combining it with the DSM. Section 4 shows how S-BPM models of cross-organisational processes can be integrated based on the DSM, illustrated using a process of applying for research funding that is executed across two organisations. Section 5 discusses the approach and gives an overview of future research directions.

II. THE DESIGN STRUCTURE MATRIX

The design structure matrix (DSM) has been developed as a compact visual aid to analysing and improving complex system architectures. It is a square $N \times N$ matrix representing the relationships between N system elements. The systems to which the DSM has been mainly applied include product designs, engineering processes and organisations [7, 5]. This Section gives a brief overview of the basics of DSM and its use for process integration.

A. Basics of DSM Representations of Processes

The DSM approach views processes as systems whose elements are activities that are interrelated by informational dependencies; i.e. one activity depends on information produced by another activity. An example of a process architecture DSM is shown in Fig. 1. The shaded cells along the diagonal represent the activities of the process, whose names are usually written to the left of the corresponding rows

In: W. Schmidt, A. Fleischmann, L. Heuser, A. Oberweis, F. Schönthaler, C. Stary, and G. Vossen (Eds.): Proceedings of the Workshop on Cross-organizational and Cross-company BPM (XOC-BPM) co-located with the 17th IEEE Conference on Business Informatics (CBI 2015), Lisbon, Portugal, July 13, 2015.

(and sometimes above the corresponding columns). Marks in the off-diagonal cells indicate the existence of an informational dependency between two activities. In Fig. 1, reading along a row indicates the outputs of an activity, i.e. the activities to which information is provided. For example, activity D provides information to activities F and G. In turn, reading down a column reveals an activity's inputs, i.e. the activities providing required information. For example, activity F depends on information provided by activities C, D and E. It should be noted that many examples from the DSM literature also use the opposite convention, i.e. representing inputs in the columns and outputs in the rows. The DSM research community has not agreed on a uniform convention to date.

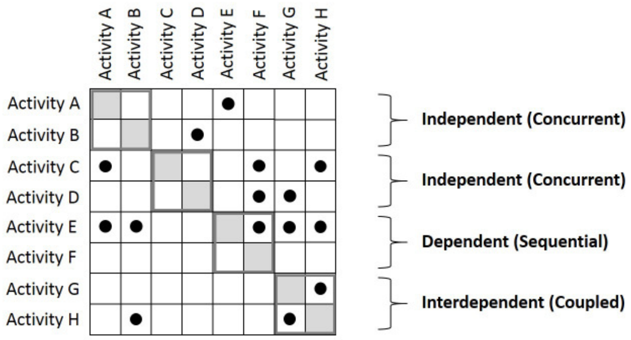

Fig. 1. Example of a DSM (adapted from [5]).

The DSM in Fig. 1 is a binary DSM because the marks merely indicate the presence or absence of a relationship. There are also numerical DSMs, where decimal numbers replace the binary marks to indicate the "strength" of a relationship, for example the likelihood or frequency of interaction between the activities. In addition, the duration of activities may be written in the empty fields on the diagonal. While numerical DSMs allow for sophisticated types of process analysis, this paper will focus on binary DSMs that provide a sufficient basis for understanding activity relationships and identifying iterations. Fig. 1 illustrates different types of relationships between adjacent activities (highlighted by squares encompassing groups of four cells along the diagonal):

- Independent (concurrent) activities: There is no interaction between activities (A and B, and C and D in Fig. 1); they can thus be executed concurrently (i.e. in parallel).
- Dependent (sequential) activities: Only a one-way relationship exists between activities (activities E and F in Fig. 1), leading to their sequential execution (potentially with some partial overlapping).
- Interdependent (coupled) activities: The activities are connected by subdiagonal and superdiagonal marks (G and H in Fig. 1), indicating a coupling between these activities.

The sequence of activities as they occur in a process is represented by their ordering from top to bottom (and left to right) in a DSM. As a result, iterations can be identified by marks that are located below the diagonal. For example, (the downstream) activity H provides feedback to (the upstream) activity B, which causes iteration as it leads to activity B having to do (partial) rework. To eliminate iterations or reduce their impact, the DSM offers a range of techniques. One of them is the resequencing of activities in a way that the marks below the diagonal are shifted closer to or above the diagonal, which is also known as "partitioning" of the DSM. A number of algorithms have been developed for this purpose [3, 8, 9] and implemented in DSM analysis tools.

Fig. 2 shows the result of partitioning the DSM from Fig. 1. The iterations are now clearly reduced in both number and size, making the process more likely to run on schedule as the impact of rework is decreased.

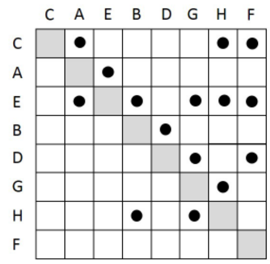

Fig. 2. Partitioned DSM (adapted from [5]).

As shown in Fig. 2, partitioning may not be able to remove all subdiagonal marks in a DSM. To address any remaining iterations, other techniques are available [5] that will be mentioned here only briefly:

- Decomposition: breaks down coupled blocks of activities into more fine-grained activities that are then partitioned again to (partially) decouple the higher-level activities.
- Aggregation: subsumes coupled blocks of activities into a single activity. While this runs into the risk of simply hiding issues, the aggregated activity can now more easily be assigned to the same organizational unit that may resolve the issues through improved teamwork.
- Tearing: replaces some informational dependencies with assumptions, leading to subdiagonal marks being temporarily removed ("torn") from the DSM. This is followed by a further cycle of partitioning. More information about tearing is provided in [3] and [5].

B. *Using the DSM for Process Integration*

Two features of the DSM make it a suitable tool for process integration: Firstly, the DSM supports all levels of granularity in the representation of processes, including detailed task structures of individual processes and high-level architectures of interlinked processes such as cross-organisational processes. Secondly, the DSM supports modular representations that separate specific process parts such as the individual processes of a (cross-organisational) process network. This Section explains the use of these DSM features for process integration, based on an example presented in [10].

The DSM in Fig. 3 is a general representation of the relationships between three processes carried out by different organisations. Note that the elements of the DSM are now complete processes rather than individual activities of a process. In addition, there are extensions above and to the right of the matrix; they represent inputs from and outputs to external processes. Reading along the extended column "A" reveals that process A receives an input from external processes, and reading along the extended row "A" shows that process A also provides an output to external processes. In the example in Fig. 3, all three processes receive input and provide output to external processes. They are also strongly coupled among one another, as shown by the marks in the main matrix.

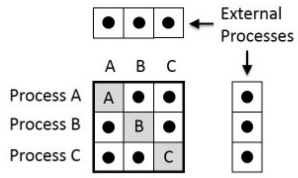

Fig. 3. DSM of interconnected processes (adapted from [10]).

The strong coupling of the processes indicates that there is potential for improving the overall process. However, reducing the coupling by resequencing the processes is impossible. Integration here needs to be done on a more detailed level, by decomposing each of the processes into more specific activity structures. The DSMs of the individual processes A, B and C are shown in Figures 4, 5 and 6, respectively.

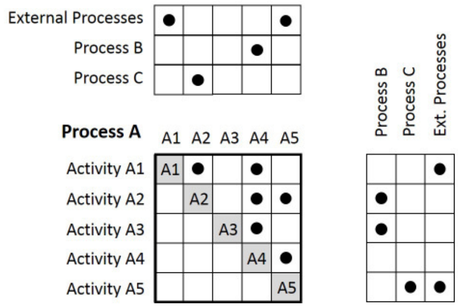

Fig. 4. DSM of process A (adapted from [10]).

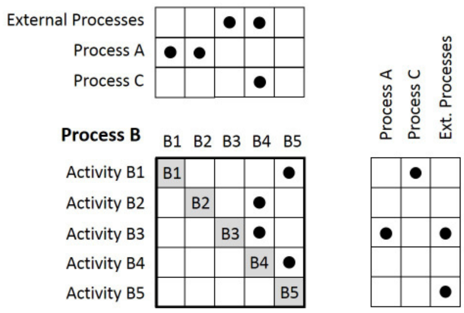

Fig. 5. DSM of process B (adapted from [10]).

The DSMs show that each individual process looks reasonably well structured, with most of the dependencies between activities being feedforward (superdiagonal). However, putting the individual DSMs together in an overall DSM, as shown in Fig. 7, reveals that iterations occur across the three processes.

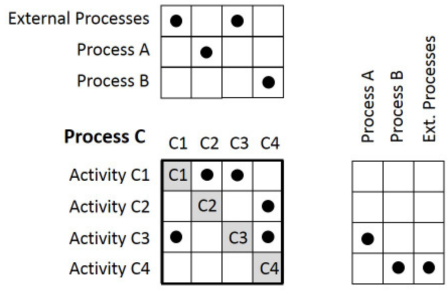

Fig. 6. DSM of process C (adapted from [10]).

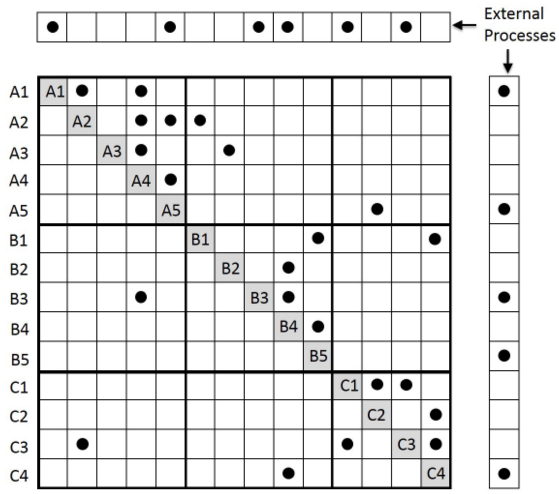

Fig. 7. Decomposed DSM of the interconnected processes (adapted from [10]).

The decomposed DSM can now be partitioned by resequencing the individual activities, resulting in the modified DSM shown in Fig. 8. The iterations are now minimised, leading to better alignment across the different processes and potentially a smoother execution of the overall process.

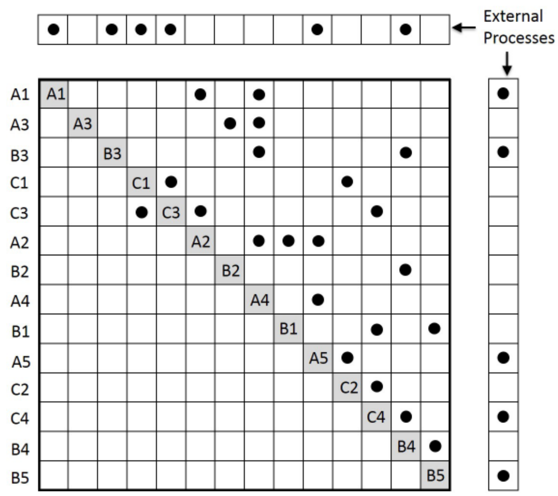

Fig. 8. Decomposed DSM of the interconnected processes after partitioning (adapted from [10]).

III. SUBJECT-ORIENTED BUSINESS PROCESS MANAGEMENT

Subject-oriented Business Process Management (S-BPM) [6] was first proposed by Albert Fleischmann in 1994 [11]. It is used in a number of organisations and companies varying in size from 500 up to 100,000 employees, including FI-TS [12], NEC [13] and Swisscom [14]. The S-BPM approach differs from traditional process modelling methods in that it is based on a decentralised view: Processes are understood as interactions between process-centric roles (called "subjects"), where every subject encapsulates its own behaviour specification [15]. Subjects coordinate their individual behaviours by exchanging messages. Such a communication-based approach differs from traditional BPM paradigms that require the orchestration of activities via tokens being passed along a central control flow. Messages in S-BPM may include information at any level of granularity, from simple notifications or requests to complex data structures (referred to as business objects).

S-BPM models include two types of diagrams: A Subject Interaction Diagram (SID) specifying a set of subjects and the messages exchanged between them, and a Subject Behaviour Diagram (SBD) for every subject specifying the details of its behaviour. SBDs specify subject behaviour using state machines, in which every state represents an action. There are three types of states in S-BPM: "receive" states for receiving messages, "send" states for sending messages, and "function" states for performing actions operating on business objects (i.e., actions performed without involving other subjects). The symbols used in the two diagrams are explained in Fig. 9.

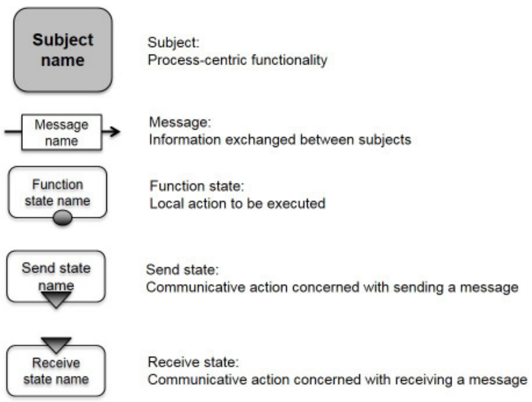

Fig. 9. The principal notational elements used in S-BPM.

An example of a SID is shown in Fig. 10, describing the subjects involved in an ordering process and the messages they exchange. Note that the "Shipment" subject is marked as an "external" subject, which means that it is an interface to another process linked to the ordering process. In the example, this linked process could be termed "supply process". It may contain further subjects (e.g. "Production"); however, they are not visible for the ordering process as they do not directly interact with it. The SID of the supply process is shown in Fig. 11. From the perspective of this process, "Shipment" is now considered an internal subject, interacting with the external subjects "Order handling" and "Customer".

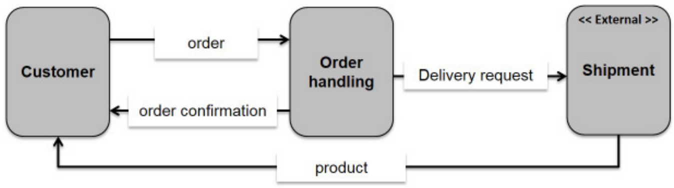

Fig. 10. Subject Interaction Diagram (SID) of an ordering process.

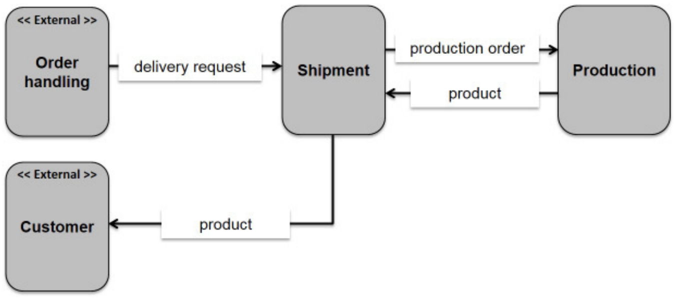

Fig. 11. Subject Interaction Diagram (SID) of a supply process.

The distinction between external and internal subjects is important because only internal subjects have visible behaviour specifications (SBDs) in the process. In contrast, external subjects exhibit merely a "black-box" behaviour in terms of the messages they receive and send. In the ordering process, only the (internal) "Customer" and "Order handling" subjects have SBDs that are shown in Fig. 12. The Figure highlights the pairs of "send" and "receive" states that establish a particular message exchange defined in the SID.

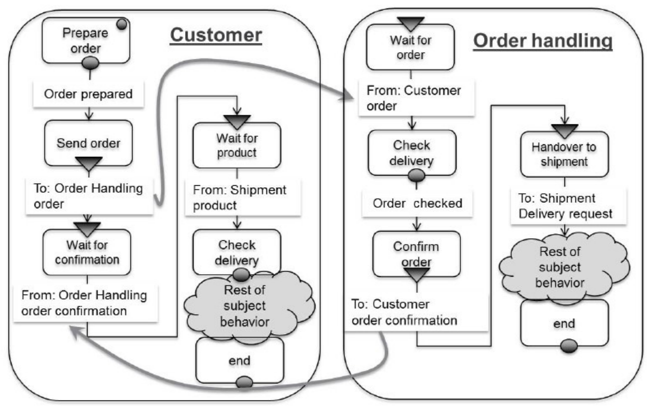

Fig. 12. Subject Behavior Diagrams (SBDs) of "Customer" and "Order handling" subjects.

One of the key features of S-BPM is its support for asynchronous message exchange. It is based on the input pool concept that can be viewed as a mailbox for all incoming messages. Every subject has such an input pool. It can be illustrated using the SBD for the "Customer" subject in Fig. 12. When this subject is in the "receive" state "Wait for confirmation", it can access its input pool and check for messages of type "order confirmation". As long as there is no such message in the input pool, the subject remains in the receive state. When the message "order confirmation" arrives (from "Order handling"), the "Customer" subject removes that message from the input pool and follows the transition to the next function state defined in the SBD. The input pool can be structured according to behaviour options: The process modeller can define how many messages of which type and/or from which sender can be deposited and what the reaction is if these restrictions are violated. In most cases, input pools are specified without any of these restrictions, allowing for asynchronous communication and thus higher concurrency of the activities executed by different subjects. If a message exchange must occur synchronously, the modeller needs to set the maximum size of the input pool to zero [6].

The asynchronicity of subject interactions is a differentiating feature of S-BPM with respect to other process notations where the orchestration of activities is synchronous (via tokens passed along control flows). An implication of this is that the sequence in which subjects become "active" (i.e. start executing their individual behaviours) is not deterministic.

IV. Process Integration Using S-BPM and DSM

A. S-BPM and DSM: How They Relate to Each Other

The explicit representation of the information exchanged as messages between subjects in SIDs allows a direct mapping between S-BPM models and DSMs. The rows and columns of a DSM can be filled with the names of all internal subjects in the SID. Where there is a message between two internal subjects, the corresponding cell in the DSM is filled with a mark. Any external subject in the SID is represented using additional tables above (if that subject provides information) and/or on the right-hand side (if that subject receives information) of the DSM, as shown in Fig. 13.

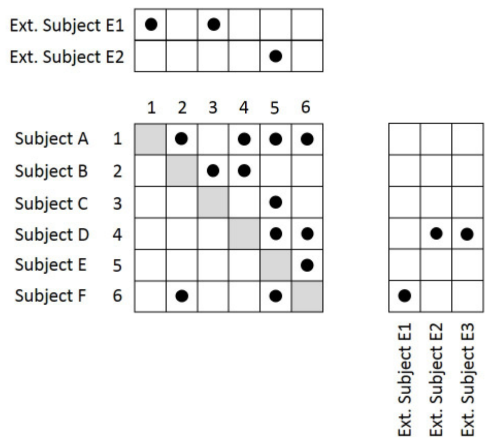

Fig. 13. Subject interactions transformed into a DSM.

Yet, the ordering of the subjects in the DSM is not obvious due to the non-deterministic execution of different subjects (see Section III). While one may expect a likely sequence of subject executions that may be called the "happy path", there is generally no guarantee for any specific sequence unless the process modeller synchronises subject behaviours based on input pool configuration or specific coordination messages.

To find the most likely sequence of subject executions, the S-BPM model must be validated [6], by simulating the execution of the process as a role play. Here, the process is "played through" by stakeholders in an offline environment, focussing on the semantic correctness of the business logic and the exchange of messages. When using only "happy-path" assumptions and decisions during such a validation session, the sequence in which subjects are executed can be seen as an indication for an expected ordering of these subjects. Commercial software support for executing and documenting validation runs of S-BPM models is available (www.metasonic.de/en/metasonic-proof).

B. S-BPM/DSM Based Process Integration: An Example

The application of a combined S-BPM/DSM approach to process integration can be illustrated using a cross-organisational research funding process involving four partial processes: (1) a funding application process executed by an SME aiming to apply for external research funding, (2) an application support processes executed by a regional government organisation devoted to assisting local SMEs in the application process, (3) a partner proposals process executed by a number of potential research partners, and (4) a funding decision process executed by a research funding body. The four processes and their relationships are represented using the DSM depicted in Fig. 14. It shows that the scope for process integration includes only the two processes "funding application" and "application support" (as they are in the main matrix). "Funding decision" and "partner proposals" are considered as external inputs and outputs; in the example they are not intended to be integrated with the other two processes.

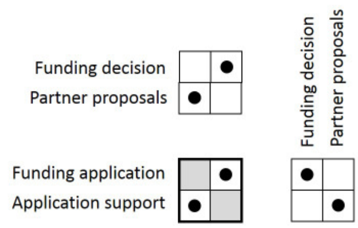

Fig. 14. Top-level DSM of the research funding process.

The S-BPM model[1] of the overall cross-organisational process comprises a SID for "funding application" (Fig. 15) and a SID for "application support" (Fig. 16). Efforts were made to lay out the two SIDs in a way to show the expected sequence of subject executions from left to right. It can be seen that there are almost no coupled relationships among the internal subjects in each diagram, which is confirmed by the corresponding DSMs shown in Fig. 17 (for "funding application") and Fig. 18 (for "application support").

The decomposed DSM in Fig. 19 of the research funding process shows the possible iterations across the two partial processes. Partitioning[2] the DSM interleaves the ordering of individual activities across the two processes, reducing the iterations as shown in Fig. 20.

[1] The S-BPM diagrams in this Section were produced using the Metasonic Suite (www.metasonic.de/en).

[2] The partitioning was supported by a DSM Excel Macro developed at MIT (www.dsmweb.org/en/dsm-tools/research-tools.html)

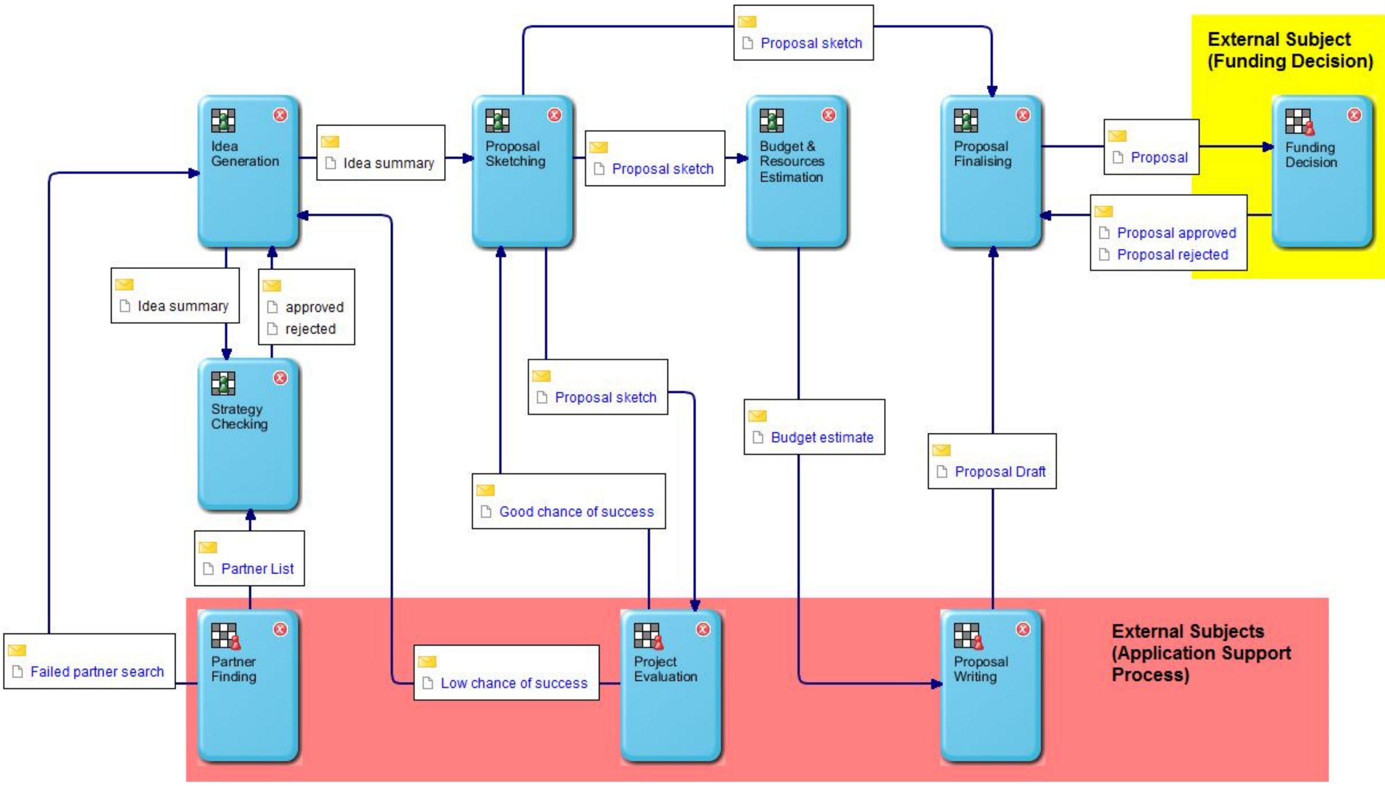

Fig. 15. Subject Interaction Diagram (SID) of the funding application process.

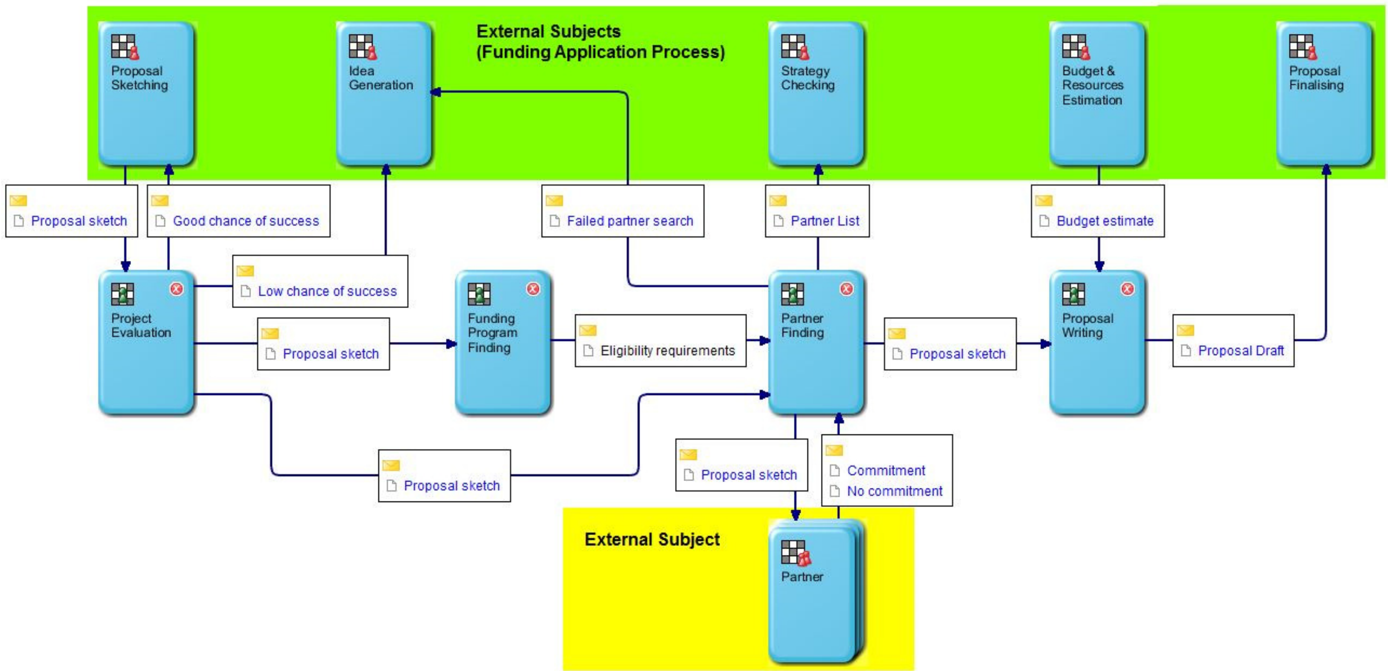

Fig. 16. Subject Interaction Diagram (SID) of the application support process.

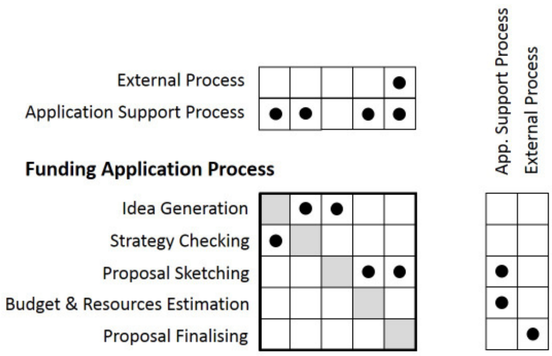

Fig. 17. DSM of the funding application process.

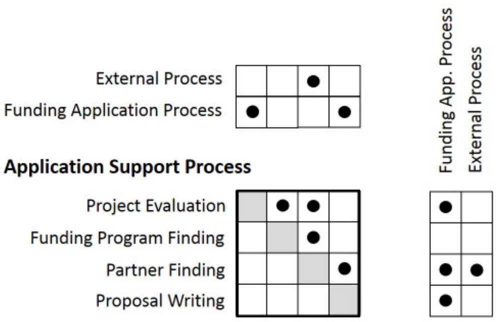

Fig. 18. DSM of the application support process.

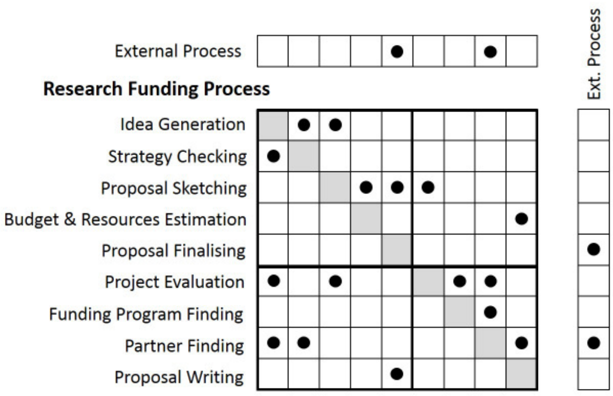

Fig. 19. Decomposed DSM of the research funding process.

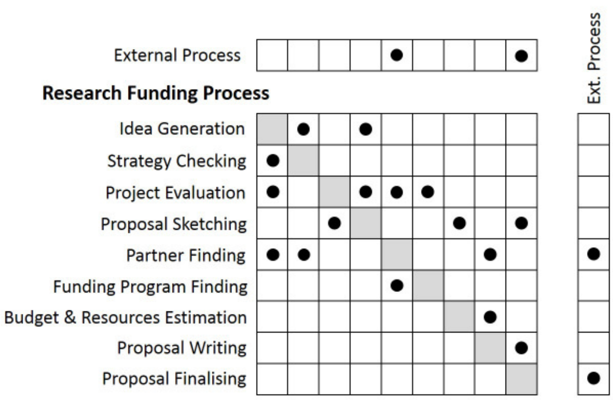

Fig. 20. Decomposed and partitioned DSM of the research funding process.

The modified DSM can now be used as a basis for checking whether the individual SBDs support the new sequence of subject executions or whether they need to be adapted. The following heuristic supports this:

1. *Bring the "send" states of a subject's behaviour in the order indicated by the corresponding row (left to right) in the modified DSM*: For example, the "send" states of the "Partner Finding" subject should be in the order: (1) Send 'Failed partner search' to "Idea Generation", (2) Send 'Partner list' to "Strategy Checking", (3) Send 'Proposal sketch' to "Proposal Writing".

2. *Bring the "receive" states of the SBD in the order indicated by the corresponding column (top to bottom) in the modified DSM*: For example, the "receive" states of the "Partner Finding" subject should be in the order: (1) Receive 'Proposal sketch' from "Project Evaluation", (2) Receive 'Eligibility requirements' from "Funding Program Finding".

3. *Arrange the "receive" and "send" states to match the business logic, add function states where appropriate*: This involves pairing up specific input and output messages and bringing them in a logical order, e.g. receiving the proposal sketch before (finding suitable partners according to the proposal sketch and then) sending off a partner list.

The result of applying this heuristic to the SBD of the "Partner Finding" subject is shown in Fig. 21. There may be other cases where such a simple heuristic does not suffice. For example, according to the modified DSM in Fig. 20 the "Project Evaluation" subject is executed before the "Proposal Sketching" subject. This poses a problem, because according to the SID in Fig. 16 "Project Evaluation" cannot start before it receives the "Proposal sketch" message from the other subject. This conflict can be solved in one of two ways:

1. Adapting the modified DSM by swapping the two subjects, so that "Proposal Sketching" is executed before "Project Evaluation": This would resolve the logical problem, but would also increase the feedback loop from "Project Evaluation" to "Idea Generation".

2. Introducing a new message to the SID to act as a trigger for the "Project Evaluation" subject: One solution may be to let the "Idea Generation" subject send the "Idea summary" message not only to "Strategy Checking" and "Proposal Sketching" but also to "Project Evaluation". This way "Project Evaluation" may already start executing some of its behaviour even though some important information about the project idea may come only after "Proposal Sketching" produces more details about the initial project idea. The additional message would need to be added to the DSM. This particular change of the DSM would not require another cycle of partitioning because the new mark corresponding to the new message would be located above the diagonal. In case a new message is defined from a downstream subject (i.e. leading to a new subdiagonal mark in the DSM), further partitioning is needed.

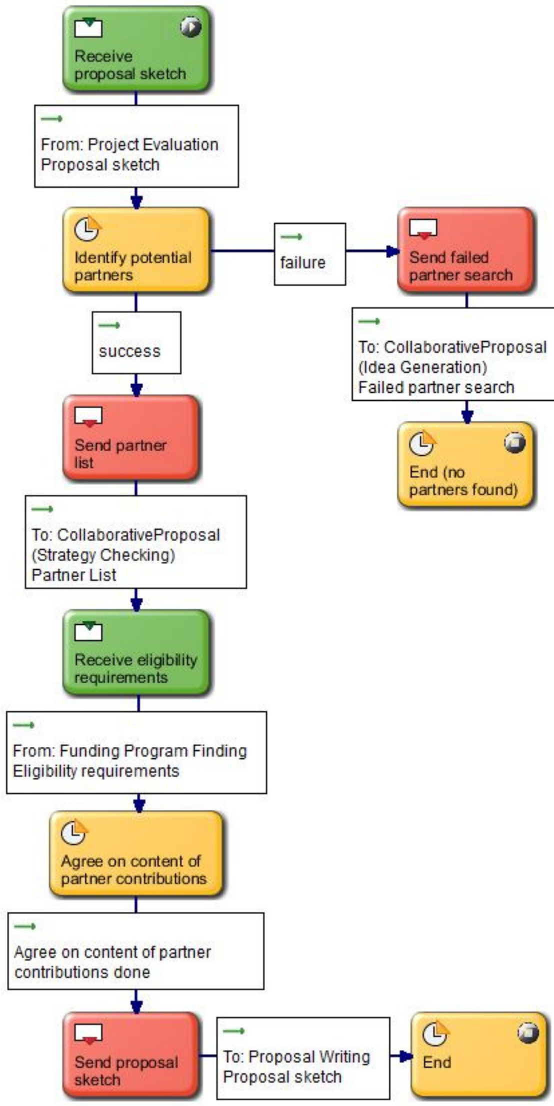

Fig. 21. SBD of the "Partner Finding" subject.

V. CONCLUSION

Cross-organisational process integration involves the decomposition of processes into interrelated activities, and their subsequent reorganisation to minimise coordination effort. The DSM is a generic tool for visualising and analysing complex (process) structures that offers these capabilities. While the DSM is well-known in the domain of engineering processes, there is almost no work on its use in BPM. This paper has shown that the S-BPM approach can open the door for DSM techniques for business process improvement including across organisational boundaries.

In today's distributed world of value creation where processes need to be integrated not only horizontally across organisations but also vertically across enterprise domains [16, 17], there is an increasing demand for generic integration tools such as the S-BPM/DSM approach presented in this paper. Future work will explore this approach for integration problems in multidisciplinary process engineering (e.g. of industrial control systems) where processes need to be integrated across different knowledge domains.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7-2013-NMP-ICT-FOF(RTD) under grant agreement n° 609190. www.so-pc-pro.eu

REFERENCES

[1] V. Allee, The Future of Knowledge: Increasing Prosperity through Value Networks, Burlington: Elsevier Science, 2003.

[2] N. Berente, B. Vandenbosch and B. Aubert, "Information flows and business process integration," Business Process Management Journal, vol. 15(1), pp. 119-141, 2009.

[3] D.V. Steward, Systems Analysis and Management: Structure, Strategy, and Design. New York: Petrocelli Books, 1981.

[4] S.D. Eppinger, D.E. Whitney, R.P. Smith and D.A. Gebala, "A model-based method for organizing tasks in product development," Research in Engineering Design, vol. 6, pp. 1-13, 1994.

[5] S.D. Eppinger and T.R. Browning, Design Structure Matrix Methods and Applications. Cambridge: MIT Press, 2012.

[6] A. Fleischmann, W. Schmidt, C. Stary, S. Obermeier and E. Börger, Subject-Oriented Business Process Management. Berlin: Springer, 2012.

[7] T.R. Browning, "Applying the design structure matrix to systems decomposition and integration problems: A review and new directions," IEEE Transactions on Engineering Management, vol. 48(3), pp. 292-306, 2001.

[8] D.A. Gebala and S.D. Eppinger, "Methods for analyzing design procedures," Proceedings ASME Third International Conference on Design Theory and Methodology, pp. 227-233, 1991.

[9] A. Kusiak and J. Wang, "Efficient organizing of design activities," International Journal of Production Research, vol. 31(4), pp. 753-769, 1993.

[10] T.R. Browning, "Process integration using the design structure matrix," Systems Engineering, vol. 5(3), pp. 180-193, 2002.

[11] A. Fleischmann, Distributed Systems – Software Design & Implementation. Berlin: Springer, 1994.

[12] G. Konjack, "Case study: AST order control processing," S-BPM ONE - Setting the Stage for Subject-Oriented Business Process Management, CCIS 85. Berlin: Springer, pp. 115-120, 2010.

[13] S. Nakamura et al., "CGAA/EES at NEC Corporation, powered by S-BPM: The subject-oriented BPM development technique using top-down approach," S-BPM ONE - Learning by Doing - Doing by Learning, CCIS 213. Berlin: Springer, pp. 215-231, 2011.

[14] T. Walke, M. Witschi and M. Reiner, "Case study @ Swisscom (Schweiz) AG: iPhone 5 self-service order app and process-workflows," S-BPM ONE - Running Processes, CCIS 360. Berlin: Springer, pp. 264-273, 2013.

[15] A. Fleischmann, U. Kannengiesser, W. Schmidt and C. Stary, "Subject-oriented modeling and execution of multi-agent business processes," 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), Atlanta, GA, pp. 138-145, November 2013.

[16] U. Kannengiesser and H. Müller, "Towards agent-based smart factories: A subject-oriented modeling approach," 2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT), Atlanta, GA, pp. 83-86, November 2013.

[17] M. Neubauer, C. Stary and F. Krenn, "Subject-oriented process design across organizational control layers," 2014 IEEE International Conference on Industrial Informatics, Porto Alegre, Brazil, pp. 418-423.