Understanding and Modelling Organisational Information Flows

Nguyen Hoang Thuan[1, 3], David Swann[2], Yi-Te Chiu[3], Pedro Antunes[3]
[1] Faculty of Information Technology, Can Tho University of Technology, Vietnam
[2] LINQ Ltd, New Zealand
[3] School of Information Management, Victoria University of Wellington, New Zealand
nhthuan@ctuet.edu.vn, David.Swann@Linq.it, {Yi-te.Chiu,Pedro.Antunes}@vuw.ac.nz

***Abstract*— Every organisation needs information flows to operate. Information modelling enables the analysis, communication and optimisation of information flows. LINQ is a technique for information modelling. This paper discusses the LINQ technique for modelling organisational information flows. We illustrate the technique with a case study. Then we analyse the similarities and differences between LINQ and other well-known information modelling techniques. The paper concludes by discussing the potential applications, future directions, and limitation of the LINQ technique.**

***Keywords*— Enterprise modelling; Information flows; Information modelling; Modelling tool.**

I. INTRODUCTION

Information modelling is an approach for better understanding how organisations structure their information flows. Such an important task can be seen from different perspectives. From a decision-making perspective, flow modelling helps organisations to understand their informational dependencies, risks and required provisions [1]. It also plays an important role in architecting, designing and engineering the organisational structures, as it allows representing the deep structures and links between services, departments, clients, etc. [2, 3]. From a more operational perspective, information flows are required to coordinate all business activities and support work cooperation [4].

Several techniques have long been used for capturing different aspects of information flows. By and large, these techniques can be grouped into two main categories: function-driven and data-driven. The function-driven category aims at capturing functional structures and their dependencies. Well-known examples of function-driven techniques include BPMN (Business Process Modelling Notation), and IDEFØ (Integrated Definition Method Ø). These techniques present organisational behaviour in terms of functions, processes, actors, events, flows, and control-flow logic. While these techniques are very effective at modelling the logic of the business functions, they predominantly focus on what must be done and see information as a secondary entity. In other words, function-driven techniques mainly consider information modelling as subsidiary to process and/or function modelling.

Data-driven techniques have been widely used to analyse the information structures needed by organisations and in particular their information systems. They tend to emphasise the structural dependencies between different data entities. Classical examples include IDEF1, Entity Relationship Diagrams (ERD), and UML Class diagrams. Most of these techniques model static data structures. They also model data flows (e.g. Data Flow Diagram - DFD) but are usually too detailed to present how business operates and to allow business users to understand business value created from information transformation. Specifically, organisations face challenges in gaining insights into high-level dynamic aspects of information flows [2, 5], including how information is transferred between processes and activities; where it comes from; what channels are used to manage the information; and who maintains the information. These challenges hinder organisational decision-making.

Addressing some of these challenges, we propose a new technique called LINQ[1] to analyse and model high-level, dynamic information flows in organisations. The LINQ technique captures organisational information transformation and analyses how information carry value (or waste) from birth of information to the end of its life within and across organisations. The technique consists of 1) a set of basic building blocks for information modelling, 2) an IASP (Information-Action-System-People) analysis that identifies LINQ building blocks, and 3) a cloud-based modelling tool graphically representing the information flows. The technique has been successfully developed and is used by several organisations. It is however noted that the LINQ technique like other information modelling techniques has its own focus and warrants further evaluation and comparison with others.

The aim of this paper is twofold. First, it introduces LINQ and illustrates its use through a case study. Second, it analyses LINQ in comparison with existing modelling techniques and demonstrates its specific value. By doing so, the current paper is relevant for both practitioners and academics. The paper emphasises on the practicality of a technique that enables practitioners to analyse and model dynamic aspects of information flows [1]. From an academic perspective, it contributes to the state of the art by analysing and comparing LINQ with the existing ones. The comparison provides a critical assessment of the modelling techniques regarding to key aspects of information flows.

[1] LINQ was developed by a company based in Wellington, New Zealand. It does not have any relationship with Microsoft LINQ (Microsoft Language Integrated Query).

II. RELATED WORK

While existing modelling techniques can be used in a variety of organisational contexts, each of them has its own focus on what business and information elements are modelled. Giaglis [6] identified four main foci often adopted by modelling techniques: functions, roles, data, and behaviours. Agreeing with Giaglis, we however note that a technique may have more than one focus. Therefore, in this study we classify modelling techniques into a primary and secondary foci. The primary focus considers the element(s) that the technique pay most of attention to and describe the general purpose of the technique. The secondary focus occurs when the technique provides ways of modelling other element(s) but in more constrained ways (e.g. dependent on other elements and non-visible). As the primary and secondary foci provide a basis for understanding a modelling technique, we use them to review a set of common techniques, including IDEFØ, BPMN, UML activity diagrams (Activity diagrams), ERD, and IDEF1.

IDEFØ, BPMN, and Activity diagrams all have their primary foci on activities and secondary foci on information. Take IDEFØ for example, as the syntax shown in Fig. 1, activities serve as the primary modelling element and connect other secondary elements like inputs, outputs, controls and mechanisms. Clearly, IDEFØ is capable of modelling information flows, but they appear as subsidiary to activities. Information flows are modelled mainly with the purpose to better understand the business activities.

In the same vein, BPMN also exhibits primary and secondary foci. Regarding the former, BPMN was developed with the purpose to model business processes. It is possible to describe information flows in BPMN, but not independently from the control flows that govern a business process. As explained by the BPMN 2.0 standard, "while BPMN shows the flow of data, and the association of data artifacts to activities, it is not a data flow language" [7]. We thus suggest that information modelling is clearly secondary in BPMN modelling. The same suggestion can be made for Activity diagrams, which also use information to support flows of activities [8].

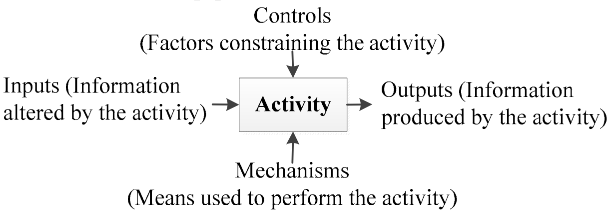

Fig. 1. Basic IDEF0 syntax [9]

Widely used as information (data) modelling techniques, ERD and IDEF1 have their primary foci on data structures. In particular, ER diagrams are network models that show the properties of data entities and their interrelationships [10]. Such emphasis on structures is reinforced by noting that ERD can be easily used to specify data structures on relational databases. Similarly, IDEF1 has the same focus. The technique models data structures, which can be handled by both computers and humans [6]. With these primary foci on data structures, the two techniques are very strong in modelling static data structures, independent of the type of processing that may take place on them [6]. However, this prevents the techniques from capturing dynamic details, particularly information flows.

A DFD depicts the data flows linking three types of elements: external entities, internal activities, and data storages [10]. This indicates a primary focus on data flows and a secondary focus on the other two elements: source or receiver of data and data store. While DFD provides a good starting point for modelling information flows, the focus on how processes connect through data stores hinders business users to understand how the information aligns with business process and creates value to organisations.

Although, over the years, numerous information modelling techniques have been developed to support system development and business (re)design, many of these modelling techniques can be highly complex (e.g., BPMN), and, in our view, business users often expressed the need to move beyond detailed descriptions to more business value-driven modelling. Given this discussion, we maintain that a gap exists in the existing modelling languages to capture the dynamic aspects of information flows. Such a technique should consist of simple notations that help business users easily analyse and understand systems and/or organisations, and maintain balance between activities and information. The technique should emphasise the information transformations, as information moves between activities, as well as the dependencies between data and activities in terms of pre and post conditions. Together, these elements would enable a model to be primarily centred on information flows. Such a model provides a high-level 'information map', which may help organisations to identify how key pieces of information relate to business outcomes and thus communicate the purpose of information flows. In short, we need a technique that is similar to BPMN in its dynamism but swap and rebalance its foci, primarily focusing on information and secondarily focusing on activities to achieve a better balance between information and activities.

III. LINQ TECHNIQUE

This section introduces LINQ, a technique that models the dynamic aspects of information flows in organisations. For this purpose, LINQ models not only information but also actions (activities), systems, people (actors), and the information transformation among them. Fig. 2 presents basic elements of LINQ.

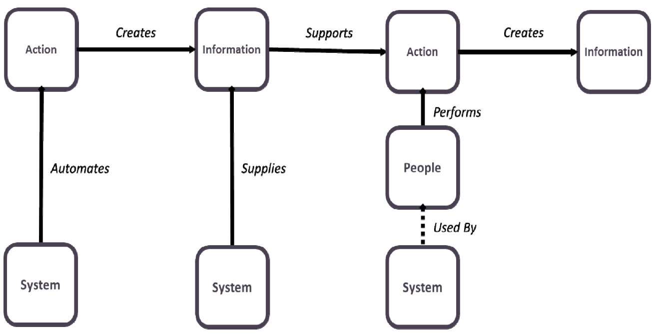

Fig. 2. LINQ basic model

At the heart of the model, LINQ defines *flows of information* as information transformations done with support from actions and in support to actions. That is, *Actions* create *Information* which supports *Actions* which create *Information* and so on, which supports a flow of information from source to business outcome. This flow of information is enabled by *People* performing *Actions;* by *Systems* being used by *People* to perform *Actions;* by *Systems* automating *Actions;* and by *Systems* providing *Information.* Together, information, actions, people, and systems are the basic building blocks defining the LINQ information flows. Within these building blocks, the primary focus is on *Information*. Based on this basic model, LINQ further defines and decomposes other building blocks, as presented below.

- Information entities can be presented in five formats: *paper, file, data-base, service, and application*.
- Action is defined as an activity or entity that creates or is supported by information. Action can be differentiated into *Capture* that creates the original source of information; *Outcome* that generates a result; and *Process* that combines, extracts, separates and/or transforms information.
- System is defined as an IT/IS implementation that supplies or processes information. It can be used in conjunction with People in order to perform an action that creates information. Systems can assume three different forms: *normal (offline) systems, on premise*, and *cloud-based* systems.
- People perform actions to create information. The building block can represent *individuals* and *groups*.

We note that to identify these building blocks in an organisational context is not always an easy task as it requires the analyst to in-deep understand the context. To support this task, LINQ provides an analysis to examine particular contexts and identify the main building blocks according the basic model. It starts with all available documentation describing the context. It then analyses the texts for identifying the main building blocks of LINQ. The analysis is best performed by colouring appropriate texts in the documents. The colours include green for Information; orange for Actions, blue for Systems, and violet for *People*. This analysis is named IASP and will be further clarified in the case study below.

A. *LINQ cloud-based tool*

A cloud-based tool was developed to implement LINQ model using an underlying node-edge graph structure. Using the web technology, the tool provides a set of graphical notations to represent the LINQ building blocks. It also offers connections among the building blocks, which enable us to represent organisational information flows.

Fig. 3 shows the tool user interface, which is organised in three areas. Area 1 represents the LINQ building blocks, each presented as a LINQ node. Area 2 is the main working canvas where information flows are drawn. This area supports drag-and-drop functions for drawing and connecting LINQ nodes. Further, to enable both abstract and detailed views on the information flows, the area allows compressing and expanding the diagrams with different levels of details. Finally, whenever a node or connection is drawn in area 2, area 3 defines its properties, keeps track on navigation, and enables the node search. The tool is available at http://www.linq.it/ for online use.

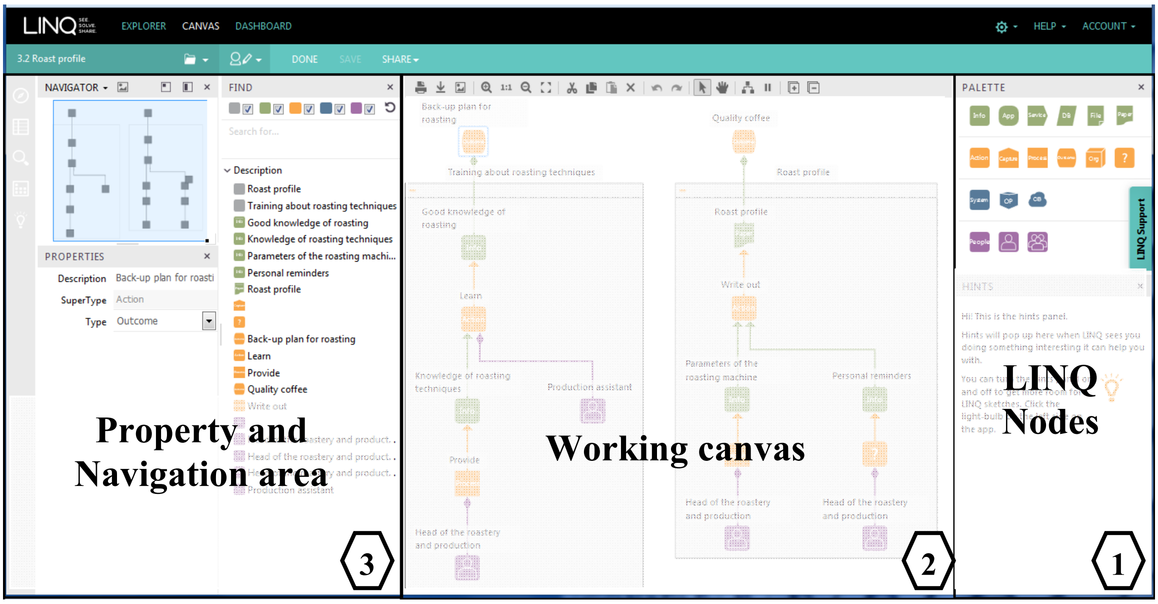

Fig. 3. LINQ user interface

IV. CASE STUDY

In this section we illustrate the LINQ technique by applying it to a case study, Coffee Co [11]. After an overview of the case, we show how to use the IASP analysis and the LINQ tool for modelling information flows of the case.

A. *Overview of the case – Coffee production plan*

Coffee Co was established in 2001, as a specialty coffee company with 15 cafés across Wellington. The company is accelerating the expansion pace of its national retail footprint. Thus, well-coordinated information flows within the organisation can increase its efficiency while reducing the problems caused by the expansion. For this purpose, the organisation adopted LINQ for modelling its information flows. Given the limited space of the paper, we focus only on 'coffee production plan'. Details are presented by the Coffee Production Manager, Nancy Maydee, through a video in this link: Video link.

B. *IASP analysis*

The IASP analysis was applied to the transcribed texts of the 'coffee production plan' in order to identify the LINQ building blocks. The appropriated texts were coloured (green for Information; orange for Actions, blue for Systems, and violet for *People*). The analysis results are presented below[2].

My days begin early. First thing I do when I arrive at the roastery is to turn on our Ghibli R15 and our Probat-Probatino, our roasting machines. The machines need at least an hour to warm-up. I use the Ghibli to roast batches up to 15 kg and I use the Probat to roast small batches of 1 kg. [This paragraph mainly includes physical actions].

I [Nancy Maydee] then check the production plan for the day. The production plan is basically an Excel spreadsheet compiled the previous evening by Paul, our wholesale manager. The production plan contains a list of coffee batches to be roasted, the final weight that has been ordered, and our inventory to date for each batch. Paul adds up both the purchase orders from our cafés and from our wholesale customers when drafting the production plan. We work this way because some customers order different quantities of the same coffee batch on the same day; it wouldn't be economical to do separate roasts of the same coffee for each customer. Paul just adds up the orders and I do a single roast for each coffee batch.

Our green coffee batches are of two sorts. They can be single lot: a specific varietal from a specific producer, or what we call a "single origin" coffee. For instance, we have just received yesterday a lot of ten 60 kg bags of Ethiopian Konga. Our batches can also consist of pre-roast blends. These blends combine beans from multiple lots, in order to achieve the flavour profile. For instance, a 15-kg batch of our "WakeUp" blend combines 5 kg from a Columbian lot, 7 kg from a Brazilian lot, and 3 kg from a Kenyan lot. [Some candidate categories of information can be found, yet they need to be further confirmed].

We note that texts in the brackets "[" and "]" are notes from the modeller to further clarify the transcribed texts. Further, it is not always clear identification of LINQ building blocks, and the modeller may change the identified ones when she/he restudies the case. Thus, it is advisable to cross-check and/or re-check the analysis results.

[2] This section consists of colour texts.

C. *Information flow diagram*

With the building blocks identified by the IASP analysis, we are now able to draw the information flows of the case. We did that by first drawing the information nodes. This step focused on *information nodes* that are transformed by actions or stored by systems: the production plan, a list of coffee batches, inventory to date, the final order, whole sale order, and café order (presented in the texts as 'purchase orders from café and wholesale'). In contrast, information that is not actually involved in the information flows was eliminated. For instance, although the analysis identified several candidate pieces of information in the last paragraph, they did not actually transform and/or connect with other elements, and thus were eliminated in the diagram.

After drawing the information nodes, we asked what actions create or use the information. Answering the questions enabled us to add *action nodes* and links them with the information nodes. Finally, we drew the identified *people and system nodes* that provide supports for action and information nodes. As a result, Fig. 4 presents a diagram modelling information flows of the case.

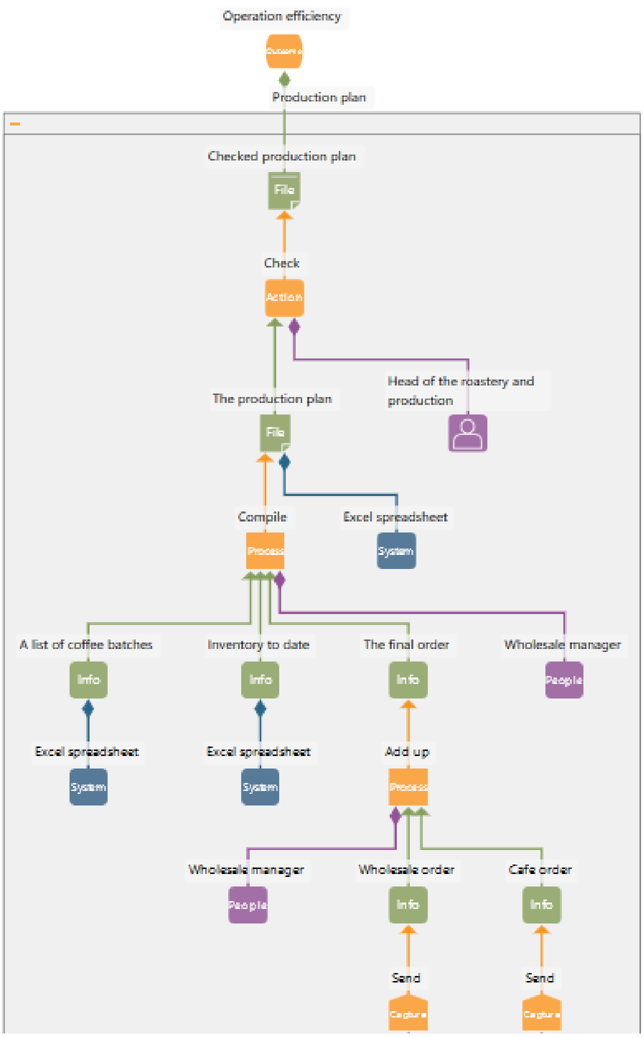

Fig. 4. Information flows of coffee production plan

Fig. 4 shows how information has been coordinated and transformed in the 'coffee production plan'. Consider, for instance, the order processing that is presented in the lower part of Fig. 4. In this part, the organisation first captures the wholesale order and café order, which are together added up to become the final order by the wholesale manager. The figure also shows how the detailed information is transformed. For example, the production plan is checked to become a valid production plan. Although the two pieces of information may not be largely different in their data concepts, their meanings and value are different, which is very important to the organisational operations.

In a broader view, Fig. 4 presents an information map of the 'coffee production plan'. Through this map, we can understand where the information comes from, where it starts and ends, who creates it, what systems manages the information, and more importantly the coordination among information, actions, IT systems, and actors. In short, the dynamic aspects of information flows are modelled.

V. COMPARATIVE ANALYSIS

The two previous sections introduced the LINQ technique. This section compares it with the common information modelling techniques, which provides a guide to practitioners and academics to understand, compare, and decide what technique to be used. In this comparison, we analyse the modelling techniques regarding their goal, focus, and main modelling elements.

It is important to identify goals of modelling techniques, since different techniques are largely suitable for different purposes [12, 13]. Phalp et al. [13] categorise two purposes of the modelling techniques: software development and business process modelling. We categorise ERD, IDEF1, and DFD into the software development category as they are widely used for IS development. We extend the second category, which includes not only business process modelling e.g. BPMN and Activity diagram, but also function modelling e.g. IDEFØ, and information modelling e.g. LINQ.

Another criterion to compare among the modelling techniques is its analysis focus. The criterion was already discussed in Section 2 as each modelling technique has a primary focus and may also have a secondary focus.

The final criterion we take into account considers the main modelling elements used by these techniques. Aligning with Rosenkranz and Holten [2], we check whether the technique includes the following modelling notations: information element, information attribute, information flow, activity (action), control flow, and actor. There are three outcomes from this check. If the technique describes and operationalises the notation, the outcome is 'yes'. If the technique just mentions the notation without its operationalisation, the outcome is 'partial'. Otherwise, the technique does not include the notation ('no').

Using the defined criteria, we compare BPMN, Activity diagrams, IDEFØ, ERD, IDEF1, DFD, and LINQ. The results of this comparison are presented in Table. 1.

TABLE I. COMPARATIVE ANALYSIS OF INFORMATION MODELLING TECHNIQUES

Modelling Technique	Description	Goal	Focus		Notational Elements						Ref(s)
			Primary	*Secondary*	*Information elements*	*Information attributes*	*Information flows*	*Activities (actions)*	*Control flows*	*Actors*	
BPMN	Provide a graphical notation to specify business processes	Business process modelling	Activity, control flow	Information, actor	Partial	No	Partial	Yes	Yes	Yes	[7]
Activity diagram	Describe flows of activities along with conditional behaviours	Business process modelling	Activity, control flow	Information, actor	Partial	No	Partial	Yes	Yes	Yes	[14]
IDEFØ	Illustrate functional modelling along with activities, constraints, and information	Function modelling	Function, activity	Information flow (input & output)	Partial	No	Partial	Yes	No	Partial	[6, 15]
ERD	Network models describing the stored data layout	Database design	Entity, data attribute, entity relationship		Yes	Yes	No	No	No	No	[6, 10]
IDEF1	Capture the data structures of the information systems	Database design	Entity, data attribute, entity relationship		Yes	Yes	No	No	No	No	[6, 15]
DFD	Analyse data flows for the design and deployment of the information systems	Software development	Input, output (static information flows)	Activity, data store	Yes	Partial	Yes; static information flows	No	No	No	[10]
LINQ	Model the dynamic aspects of information flows along with actions, systems, and actors	Information modelling	Transformed information, sequential dynamic information flows, and activity	Actor, system	Yes	No	Yes	Yes	Partial	Yes	Section 3 & 4

Overall, the comparative analysis presented in Table 1 leads to two interesting observations. First, the analysed techniques differ significantly according to their goals, focus, and modelling elements. Thus, depending on the modelling contexts, appropriate modelling techniques should be chosen. Table 1 serves as a starting point for guiding this choice.

Second, the three analysed criteria clearly show the differences between LINQ with other modelling techniques. While most of the current techniques focus on activities/processes (e.g. BPMN and IDEFØ), data structure (e.g. ERD and IDEF1), and static aspects of information flows (e.g. DFD), LINQ primarily focuses on the dynamic aspects of information flows. Further, LINQ is one of a few modelling techniques that consists of notations for modelling information flows and at the same time takes into account activities, actors, and their interactions. Consequently, we suggest that LINQ meets its purpose of modelling organisational information flows.

VI. CONCLUSION

There is a need for information modelling techniques that can capture the dynamic aspects of information flows [2, 16]. Addressing the need, this paper has introduced LINQ with its focus on information flows and information transformation among actions, actors and IT systems. A case study was used to demonstrate how the technique can be practically applied to model the information flows in the organisation. We also compared LINQ and showed its difference with other modelling techniques regarding goals, focus, and the main modelling notations.

Our research contributes to the current knowledge by proposing the LINQ technique that contains the basic building blocks for information modelling, the ISAP analysis, and the tool. Together, they enable us to model, analyse, and graphically represent information flows of the organisation. Researchers may also benefit from our comparative analysis that provides insights for choosing an appropriate modelling technique. From a more practical perspective, the LINQ tool is launched online, ready for practitioners to visit and model their information flows.

Through a critical lens, we identify two main limitations of the current work. First, although the case study shows how LINQ can be applied to model information flows, more formal evaluation should be conducted. Second, we note that LINQ presents information at a high level of abstraction. While this abstraction enables an overall picture of the information flows, it does not show the detailed data structures and thus may be limited from the perspective of software development that need the details for database design and development.

Future work should address these limitations and strengthen the LINQ technique. We invite researchers and practitioners to apply LINQ to different modelling contexts, which will allow us to further assess the strengths and weaknesses of the technique. Another interesting direction is to integrate LINQ with low-detail modelling techniques, like ERD and IDEF1. This integration would provide a more unified approach, capturing information flows and then transferring them into information systems that support these information flows. Finally, we are going to investigate how to apply the value-added analysis to the modelled information flows. Similar to the benefits of applying the value-added analysis to business processes [17], this application possibly allows us to evaluate the flows of information, identify the most important ones, remove unnecessary wastes, and thus increase efficiency of the information flows.

REFERENCES

[1] C. Durugbo, A. Tiwari, and J. R. Alcock, "Modelling information flow for organisations: A review of approaches and future challenges," *International Journal of Information Management,* vol. 33, pp. 597-610, 2013.

[2] C. Rosenkranz and R. Holten, "The variety engineering method: analyzing and designing information flows in organizations," *Information Systems and E-Business Management,* vol. 9, pp. 11-49, 2011.

[3] A. Albani and J. L. Dietz, "Enterprise ontology based development of information systems," *International Journal of Internet and Enterprise Management,* vol. 7, pp. 41-63, 2011.

[4] C. Durugbo, A. Tiwari, and J. R. Alcock, "Managing integrated information flow for delivery reliability," *Industrial Management & Data Systems,* vol. 114, pp. 628-651, 2014.

[5] N. Kock, "Communication-focused business process redesign: assessing a communication flow optimization model through an action research study at a defense contractor," *IEEE Transactions on Professional Communication,* vol. 46, pp. 35-54, 2003.

[6] G. M. Giaglis, "A taxonomy of business process modeling and information systems modeling techniques," *International Journal of Flexible Manufacturing Systems,* vol. 13, pp. 209-228, 2001.

[7] OMG, "Notation (BPMN) version 2.0," *OMG (Object Management Group) Specification,* 2011.

[8] C.-H. Kim, R. H. Weston, A. Hodgson, and K.-H. Lee, "The complementary use of IDEF and UML modelling approaches," *Computers in Industry,* vol. 50, pp. 35-56, 2003.

[9] S.-H. Kim and K.-J. Jang, "Designing performance analysis and IDEF0 for enterprise modelling in BPR," *International Journal of production economics,* vol. 76, pp. 121-133, 2002.

[10] H. J. Rosenblatt, *Systems analysis and design*: Cengage Learning, 2014.

[11] J. G. Bernard, B. A. Aubert, Y.-T. Chiu, and G. Oliver, "Coffee Co Case," Technical report, 2016.

[12] R. S. Aguilar-Saven, "Business process modelling: Review and framework," *International journal of production economics,* vol. 90, pp. 129-149, 2004.

[13] K. T. Phalp, "The CAP framework for business process modelling," *Information and Software Technology,* vol. 40, pp. 731-744, 1998.

[14] R. M. Bastos and D. D. A. Ruiz, "Extending UML activity diagram for workflow modeling in production systems," in *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, 2002, pp. 3786-3795.

[15] Knowledge Based Systems. Inc. (2016). *IDEF family of methods: a structured Approach to Enterprise Modeling & Analysis*. Available: http://www.idef.com/

[16] C. Durugbo, W. Hutabarat, A. Tiwari, and J. R. Alcock, "Information channel diagrams: an approach for modelling information flows," *Journal of Intelligent Manufacturing,* vol. 23, pp. 1959-1971, 2012.

[17] S. Conger, "Six sigma and business process management," in *Handbook on Business Process Management 1*, J. vom Brocke and M. Rosemann, Eds., ed: Springer, 2015, pp. 127-146.