



HAL
open science

Implementing an Event-Driven Enterprise Information Systems Architecture: Adoption Factors in the Example of a Micro Lending Case Study

Kavish Sookoo, Jean-Paul Van Belle, Lisa Seymour

► **To cite this version:**

Kavish Sookoo, Jean-Paul Van Belle, Lisa Seymour. Implementing an Event-Driven Enterprise Information Systems Architecture: Adoption Factors in the Example of a Micro Lending Case Study. 10th International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS), Dec 2016, Vienna, Austria. pp.293-308, 10.1007/978-3-319-49944-4_22 . hal-01630548

HAL Id: hal-01630548

<https://inria.hal.science/hal-01630548v1>

Submitted on 7 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Implementing an Event-driven Enterprise Information Systems Architecture: Adoption Factors in the Example of a Micro Lending Case Study

Kavish Sookoo, Jean-Paul Van Belle and Lisa Seymour

Department of Information Systems, University of Cape Town, Cape Town, South Africa

Jean-Paul.VanBelle@uct.ac.za

Abstract. Event-driven architecture (EDA) is an architectural approach for enabling communication between distributed enterprise information systems. EDA enables organisations to be adaptable, flexible and robust in the management of business processes and ultimately achieving agility. This paper reviews definitions, concepts and adoption criteria for an EDA and a case study investigates the adoption of the EDA in a micro lending organisation exploring the technological, organisational and environmental adoption factors.

Keywords: Event-driven architecture, distributed EIS, SOA, IS adoption

1 Introduction

Organisations need to be agile in order to accommodate changes to environmental conditions and customer demands [1]. To achieve agility, organisations are obligated to react to opportunities and pressures in order to continuously monitor and optimise business processes. In order to do this, organisations are dependent on the underlying information systems to make decisions pertaining to these business processes [2].

In the development of enterprise-wide information systems, communication, both internal and external to the organisation, is required. This communication is not only between information systems but within the components of the information system [3]. To facilitate this communication, information systems are reliant on events that are responsible for the triggering of business processes. Event-driven architecture (EDA) is an emerging information systems paradigm and architectural approach that facilitates the communication between disparate information systems [4].

The research problem is that little is known about the drivers and issues relating to the implementation of EDA in a real-world context. The goal of this research is to provide a deeper insight into the adoption of an EDA, and particularly relating to the Technological, Organisational and Environmental (TOE) factors which can drive the organisational adoption of EDA. This is achieved by exploring the adoption of an EDA within the context of a Micro Lending Organisation (MLO) in the form of a case study.

2 Literature Review

The event-driven paradigm represents a real time occurrence of an event within a domain. This event could be any internal or external activity performed and can initiate a single or multiple actions. Business events are procedures that affect the organisation's business processes. These business events are a change in state, related to the business processes. This state could be a customer interacting with a product, a change in the operational environment or a reaction to competitors [5].

EDA is a methodology that allows for the production, detection, consumption and reaction to events. These events are propagated in real time to all event generators and event consumers. Multiple event consumers may respond to an event initiated by an event generator. Events are transmitted between information systems, through decoupled service components [6]. Complex event processing (CEP) involves the evaluation of a number of combined events and performing an action on these events. These events may be related casually, temporally or spatially [7]. A common usage scenario for CEP is to respond to business anomalies, threats or opportunities.

The design of an EDA relies on the publish-subscribe pattern to enable the decoupling of components. An EDA allows asynchronous communication of events to coordinate the analysis of multiple data streams in real time [8]. EDA allows the organisation to respond in real-time to events without limitations imposed by technology. These events deliver information constantly, promoting information sharing, preventing information hiding and allowing information to be acted upon by relevant business stakeholders [9].

2.1 Why Adopt an EDA?

The rise in real time systems has increased with the advent of mobile computing [10]. Event-driven design using the publish-subscribe pattern allows for communication in the form of "push" messages [11]. This removes the need for polling of events, taking control away from lower computational devices to large event processing systems. This impact in reversing control, allows for a shorter duration of processing of events and increases the real time responsiveness of systems [9].

An important aspect of EDA is decoupling. This allows for generators and consumers of events to operate independently. Generators can produce events with consumers disconnected and consumers in turn can handle events while publishers are inactive [12]. By employing the use of a decoupled communication mechanism, EDA allows for greater flexibility, ease of configuration and reconfiguration [13]. The asynchronous model of an EDA, allows event generators and event consumers to publish or consume events without waiting for a response. This allows generators and consumers to consume and generate events without the need to wait for one another [12].

The decoupling and asynchronous nature of an EDA increases scalability by removing the dependencies between event generators and event consumers. These properties result in the methodology being well adapted to large-scale distributed environments [14].

2.2 EDA Capability

EDA has been researched for its ability to secure competitive advantage in the healthcare industry [8]. The research reviews the relationship between EDA capabilities, dynamic capability and competitive advantage in healthcare organisations. Capabilities are organisational capabilities and information technology (IT) capabilities [1]. Organisation capability is defined as the competency to coordinate the organisation's resources effectively to achieve corporate performance. IT capability is defined as the ability to mobilise and deploy IT resources in combination with organisational resources and capabilities. Organisations with high IT capability tend to outperform competitors on a number of profit and cost based performance measures [1]. The research defines EDA capability as the "ability to propagate the real-time events to all interested targets automatically and to support them to evaluate and then make decisions optionally" [8]. The EDA capability is sub-categorised into capabilities reflective of EDA design principles. Table 1, describes each of the capabilities along with the correlating design principle research. Design principles are associated with capabilities such as by adopting an EDA, asynchronous communication is realised, resulting in achieving the sensing capability of retrieving information of real time processes [8].

Table 1. EDA Capability Dimensions [8]

Dimension	Definition	Design principle
Sensing Capability	Refers to the ability of event-driven IT architecture to recognize event-triggering information and to provide managerial visibility into the business processes in real time.	Asynchronous communication; Publish-subscribe pattern
Responding Capability	Emphasizes the ability of the IT architecture to support the manager's decisions and actions at each organizational level.	Asynchronous communication; Publish-subscribe pattern
Interoperability Capability	Provides a system platform which performs the exchanging and sharing of data on two or more software components	Asynchronous communication; Decoupling
Flexibility Capability	Enables an organization to speed operational changes and promote a high degree of business agility via IT architecture modularity, compatibility and maintainability	Asynchronous communication; Decoupling

2.3 Adoption Factors for EDA

The literature has established that technical, business and human factors are relevant in the adoption of Service Oriented Architecture (SOA) [15]. Many of these factors are also valid in the context of EDA. Thus, we reviewed SOA adoption factors [16] detailed in Table 2.

Organisations adopting an EDA need to understand the requirements pertaining to the adoption before proceeding. Adoption factors of event-driven SOA include the ability to have an established and reliable hosting environment to facilitate the communication of events across multiple applications in parallel [17]. Without this, a bottleneck may occur resulting in event information delayed and inherently causing performance impediments [17].

Another adoption factor is to have an ordering mechanism for events. This is to ensure that dependent information is delivered in the correct order for critical business processes [9], [17]. The lack of standards in EDA is adversely affected by non-standardization of event messages. Adhering to a particular event standard is a key requirement in the successful implementation of an EDA [9]. Currently event messages are developed according to particular vendors as opposed to an EDA standard [17].

Other factors that are a requirement for the success in the adoption of an EDA focus around the distributed environment. The recovery from failures in applications and in the hosting environment needs to be taken into account following the adoption of an EDA [10], [17]. Failures in terms of exceptions in event flows is also difficult to diagnose and a proper strategy and monitoring tools are needed for an EDA [18].

The technical aspects related to successful adoption of an EDA or any other methodology are compounded by the factors related to human resources and skills [5]. Having the appropriate skill levels and the human resources capable of understanding an EDA is paramount to its adoption. This is particularly important in CEP applications that have several events that are required to complete an action [16].

Table 2. SOA adoption factors [16]

Adoption factor	Description
Strategy and goals	A clear overall strategy and goals are required in the initiation phase. The strategy includes business and IT to meet critical business goals related to SOA adoption.
Financial implications and benefits	The financial implication and benefits for adopting an SOA, examines the revenue streams, costs and return of investment. In increasing revenue streams, SOA may provide increased revenue or decreased expenses from increased reliability offered by SOA. In addition, the time to market new services for customers is reduced due to the reusability features of SOA. In adopting an SOA, cost such as new hardware, software and human resources needs to be established to fully assess the potential factors.
IT agility-business alignment	Aligning business processes to an SOA approach requires effort from both IT and business.
Communication	Effective communication between different departments among the bank and vendors are essential for SOA adoption success.
Culture:	A culture that facilitates SOA is vital to its adoption.
Human resources	Training current staff and introducing staff competent in SOA is necessary for SOA adoption.
Risk management	In introducing a new methodology, risk is an inherent factor. Risk management and monitoring should be on going in the adoption.

2.4 Micro Lending

Access to credit is key to economic development and improving the standard of living for many people [19] and during times of crises is a fundamental aspect of recovery [20]. Traditional financial institutions require individuals to provide collateral in obtaining a loan or credit. This is normally out of reach for the poor [21]. Hence micro lending, for many years, has been a predominant credit lending facility in the informal sector. In order to accommodate the need for credit, a digital approach was taken and

online short-term loans were established as the preferred means of credit lending. Micro lending in the form of short-term loans is a relatively new market segment. A short-term loan is a monetary credit with a term not exceeding a single month [22]. Having access to credit is important in times when natural disasters such as floods, hurricanes and tornadoes occur. In the United States those affected by natural disasters and not compensated timeously by insurance companies have utilised the services of short-term loan providers to obtain much needed financial assistance [20].

In recent years short term loan providers have gained a reputation for misdirecting credit seeking consumers and charging higher than normal interest rates and fees. This resulted in an increase in legislation and regulation by governments and credit lending bodies [22]. Short-term loan providers are required to mitigate risk and to verify customers that are applying for these loans. To enable this verification, a number of information systems both internal and external to the organisation are queried for different criteria. This includes the verification against legislation requirements and validation of personal information and income. The querying of systems occurs in real time and obtains a decision on lending in a relatively short period [22]. The criteria given above and the adoption factors presented, advocate for the use of an EDA in a micro lending organisation.

3 Research Framework and Methodology

The research asked the question of what factors are required for the adoption of an EDA within a micro lending organisation. The sub-questions include:

- What technical and environmental factors did the adoption address?
- What concerns were there in the adoption of an EDA?
- How did the adoption affect the organisation and the stakeholders?

Determining the adoption of information systems is a common research area. There are a number of frameworks, models and methodologies that aid in determining how specific variables affect the adoption of a specific technology within an organisation [23]. To investigate the factors influencing EDA adoption, the research used the TOE framework [24]. The framework identifies three contexts, namely the technology, environment and organisation contexts. TOE is regarded as a framework that provides a classification of variables rather than a fully integrated conceptual framework. To enrich the TOE framework variables from other areas such as sociological, cognitive and technology readiness need to be incorporated. This is usually done by combining the TOE framework with other theories such as Diffusion of Innovation [23]. The research used these contexts in order to investigate factors retrieved from the literature.

The research has been conducted in a post positivistic manner. Post positivistic research is a paradigm that moves from the narrow perspective of positivism, enabling the examination of real world problems. This is evident in post positivism, in that it recognises that diversity and complexity are part of all experiences [25]. The research follows the deductive approach [25], using the TOE framework in conjunction with factors derived from literature to investigate the adoption factors of EDA.

The research is in the form of a case study. The research investigates a single case in an explanatory manner. The theoretical motivation is the exploratory nature of the re-

search. A practical reason for the particular case study selected is that one of the researchers is employed by the organisation and has access to stakeholders within the organisation to perform interviews and to obtain valuable information. Permission was obtained from the micro lending organisation management to conduct the research and ethics approval from the research organisation (University).

A cross-sectional approach was taken, i.e. a snapshot of the status in the organisation in August 2015. However, some of the historical evolution and experiences were included in the case study description and interviews.

Fifteen candidates were interviewed in semi-structured interviews. These included software engineers and a team lead who actively maintains and develops the EDA solution, a software development manager, a product owner that is responsible for new product features, the head of operations, the head of marketing and executive management. The international micro lending organisation has a geographically dispersed team and as a result some candidates were interviewed in person and others were interviewed with the use of video or voice calling. The questions varied depending on the interviewee's role in the organisation and knowledge of the EDA system.

Interview responses were analysed using thematic analysis and using NVivo software. During phase one, the researcher read the results and transcripts to become familiar with the content in order to identify patterns. During phase two codes were identified and documented. During phase three codes were linked to themes. Phase four determined how the themes support the results obtained and the factors presented by the TOE framework. If these themes were incomplete, the process began from phase one until all themes were identified. Phase five determined the significance of each theme and the significance of the results obtained.

4 Case Study Description

4.1 Organisational Context

The case study was performed at a MLO which is based in South Africa but part of an international organisation. The MLO is a pioneer in the micro lending space by being the first to introduce short-term loans online. The MLO was started in 2007 and has its headquarters in a major European city. Two South Africans, one a software engineer and the other an entrepreneur, founded the MLO. The founders wanted to develop a lending model that provided short-term credit in a transparent manner and with immediate access to funds. Furthermore, the solution should provide customers with a paperless and prompt response on the credit decision.

In order to provide for the prompt decision-making, the organisation developed a fully automated decision processing technology to determine the customer's risk profile. The risk model and algorithms use data points that access a number of online credit providers using application-programming interfaces (API). Once a positive decision for lending is provided from the decision engine the funds are transferred to the customer's bank accounts.

The improvement of the decision engine and algorithms led to the expansion of the MLO to several countries. This led to a number of legislation and regulation challenges as each country has its own credit regulations. In the country of origin, the MLO was a

disruptor and at the time, there was not adequate regulations and legislation on short-term lending. The South African branch of the organisation however was subjected to a lending environment that has stringent regulations and legislation. The National Credit Act (NCA) overseen by the National Credit Regulator (NCR) governs these regulations [19].

4.2 EDA System Architecture

The architecture of the MLO is based upon the need to move from a monolithic traditional system to a distributed system. The concepts around this distributed system include the use of a service bus encompassing a messaging paradigm. A service bus is an architecture pattern that facilitates the messaging between multiple services and applications. The communication between services and applications is at the lower level and provides a means to connect predominately XML based messages [9]. The service bus utilised by the MLO is based upon NServicebus a distributed methodology that is implemented using Microsoft's .NET framework.

In order to accomplish the architecture a number of design patterns were used. The publish-subscribe pattern provides for the implementation of event generators and event consumers. Another pattern that is featured in the MLO design of the distributed system is that of the Command Query Responsibility Segregation (CQRS) pattern. The pattern aims to divide the obtaining of data and changing the state of the data in a distributed system. The benefit of this is to distinguish between the actions that alter the system and those that require information from the system [26].

The interaction of the user interface to the services of the system is via a service bus using XML. The EDA solution of the MLO is comprised of four service contexts. The payments service is responsible for the processing and collection of customer payments. The Decision service is responsible for the verification of customer details and accessing the affordability of the customer. This service interacts with credit providers and the decision engine to obtain the data points of the customer. The communication service is responsible for the communication channels to the customer. This integrates with an external marketing platform that enables communication using email, SMS and social media.

The technology utilised forms part of the current architecture that was developed in 2011. Due to the initial version of the monolithic system developed using Microsoft technologies, the decision taken was to continue using the existing technology stack. This is due to having the skills familiar with the Microsoft technologies.

The distributed enterprise software framework NServicebus uses Microsoft's .NET framework [27] and uses a point-to-point service configuration where each service subscribes to the service address or endpoint to publish or subscribe messages. NServiceBus works in conjunction with a messaging system. The messaging system that is utilised within the MLO is Microsoft Message Queueing (MSMQ). MSMQ is found on almost every version of windows operating systems and allows the communication of messages between heterogeneous networks and applications [27]. It uses an underlying data store such as Microsoft SQL Server to perform event sourcing. A major framework feature is the ability of NServicebus to manage long running processes known as Sagas. These processes are scheduled to execute for an extended duration of time and normally

involve the use of batch jobs. Sagas save the state of event messages thereby enabling the framework to provide the capabilities of fault tolerance and automatic retry [27].

The MLO makes use of cloud services such as Amazon web services for the hosting of server environments. These cloud services help in alleviating the issues with having access for a distributed team requiring resources in multiple countries. Other technologies used separate from the EDA solution is the user interface layer that is based on a PHP technology stack.

5 Adoption Factors

The discussion of the hypothesised adoption factors is organised using the main categories of the TOE framework. The interview analysis and discussion has been summarized to fit within the space requirements of the paper; a more detailed analysis is available on simple request from the authors. Indication of whether the factors are positive or negative influences are indicated by (+) and (-) respectively.

5.1 Technology Context

Relative advantage (+): Supported - This is illustrated by the MLO's transition from a monolithic system. The factors of scalability, asynchronous communication and decoupling were cited as the most important in the adoption. *"Typically when someone decides to use asynchronous technologies they're expecting scale...which is a very common in a start-up."* This complements but distinguishes it from SOA: *"Generally what we're trying to achieve is a service oriented architecture. There's three ways you can achieve that one is through designing it in an Asynchronous event driven model and the other is through a typical web service model. So they both give you a service oriented architecture but in different ways..."* This is confirmed by other technical participants indicating that scalability, decoupling and asynchronous communication as crucial benefits in comparison is to traditional monolithic systems and systems that use synchronous communication.

Another factor for an EDA as opposed to an SOA, relates to the reduction in temporal coupling: A concern in the context of the MLO is the integration with third party services such as credit providers, is that when temporal coupling occurs it will affect customers. *"Another big problem that can be exacerbated even more if you're doing a request out to the web you know from an internal system out to a third party"* In order to solve this issue and move away from the service-oriented approach the MLO chose to use an EDA that removes the temporal coupling with the publish-subscribe pattern. *"What event driven architecture try to do they try to solve that problem by reducing the necessity to have those synchronous request response... service A can just service B can publish an event when the data changes and service A can listen to that event and store a local copy. Now when service A needs that information, service A no longer has that temporal coupling"*

Perceived direct benefits (+): Supported - The benefits of the adoption of the EDA are related to the fault-tolerance and reliability of the EDA solution. This is necessary for the integration with external systems. Participants from the organisations' business units indicated that reliability was a major factor in the adoption of the distributed EDA

solution. *“You know I think that an event driven architecture you have a little more long term reliability because you can maintain things in smaller chunks...”* with the fault tolerance capability of an EDA stating that event messages are reprocessed when recovering from failure. This brings the benefit of having critical business processes related to customers, are re-executed recovering from a faulted state with little or no intervention. *“You know then the comfort in knowing that messages and things like that will be resubmitted when you come back online....that is reassuring....and a lot of things kind of resolves itself which is a big benefit from a business perspective....”* A Software Engineer statement that *“the pro to being durability of the system and things being able to really being able to go down and not ever lose any progress”*, confirms this.

Complexity (-): Supported – Asynchronous communication increases complexity for software engineers. A consensus from the data collected indicated that there is an increase in the complexity of the solution. Moving from a monolithic system to a distributed system added to the complexity. *“I wouldn’t say it’s because of event driven architecture I would say it’s more to do with a distributed service orientated architecture and the fact that you know all these distributed components around the place.”* This affects developer productivity: *“Complexity to this system makes the learning curve for new developers a lot higher. By this I mean it can make the debugging a lot more tricky, harder to see...”* A notable issue experienced by the MLO in retrieving information is a disconnection between all three systems. However, a notable comment is that the *“adoption of an architecture does not necessarily make a business more complex. I think your architecture should reflect how a business is organised”*. Complexity introduced, is offset by reliability, agility and fault tolerance.

Perceived financial costs (-): Supported – Increase in cost for more hardware and resources. An increase in operational costs is experienced in the maintenance of the EDA solution. Cost and complexity have a relationship in that due to system complexities, more developers that are skilled are required. *“... The cost goes hand in hand to the complexity because obviously it takes longer...more development hours you need a lot more skills developers in order to function properly on such a system.”* Operational costs increase because of the EDA solution require more hardware to achieve its objectives. These objectives include flexibility and reliability. *“From an operational perspective it is more expensive to run the EDA solution because it runs across more machines and more hardware. Having said that, it does mean that it’s more reliable. You cannot get more reliability and more flexibility at a cheaper cost...”*

Infrastructure (+): Supported – More personnel and hardware is required for the adoption. *“There’s no specialised hardware required but there are more of it because of the fact that there are more services spread across more systems that way...”* In terms of the infrastructure to support and maintain the EDA solution additional support staff was required: *“So yes so you would need more sys ops and maybe DevOps people as well. So yeah in short...you do need more people mostly around specialisations of DevOps with the environments.”*

Technology maturity (+): Unsupported – Frameworks and tools for an EDA adoption are still improving and are not mature due to a lack of standards. When the adoption of the EDA solution commenced, not many tools addressed the need for event messages in a distributed system. As stated above, the solution was based on the Microsoft’s technology stack. Choosing this solution (NServicebus) was due to the continuity from the

monolithic by keeping with Microsoft technologies. *“I think at the time there wasn’t a hell of a lot frameworks available. Um...I think the only really...other framework that was really considered is called Mass transit.”* However, since then new open source technologies became available and the current technology selection has greatly improved. *“The tooling I think is quite good you know there’s lot of open source tools that enable you know co-ordination across services and you know tools like Nservicebus you know they are quite solid and you know their equivalence and the units were older were quite good. So tooling I think have become quite good over the past 5 years.”*

Vendors and tools (+): Partially supported – Tools are categorised around developers, monitoring and business intelligence. The predominate technologies used as stated previously is Microsoft based. However, there is lack of tooling for advanced usage scenarios. *“The tooling in IDE needs to you know evolve to the point where this is a standard sort of feature where they you know tie these handlers and things together and allow you to navigate from you know a loosely coupled one point to the other.”* Tooling for race conditions is not yet established: *“You know if you’re trying to figure out what is the exact sequence of events that caused request to fail. If it’s due to race conditions for example those can be extremely difficult to solve and to tooling around solving race conditions is still poor.”*

5.2 Organisational Context

Human resource factors (+): Supported – Adequate personnel with high level of experience is required when adopting an EDA. *“From a hiring perspective you know we do need people that are a little bit more capable and you know have stronger skills than say a simpler system.”* However, this brings advantages to the MLO as the personnel hired have a greater skillset and quality: *“I don’t think that’s necessarily a bad thing for the other advantages that brings you know in terms of just having higher quality people but it does mean it requires a different skillset.”* In terms of a specific skillset required by software engineers, a number of participants advocated for the ability and willingness to learn: *“In general developer skills would mean willing to easily learn or ability to easily learn and quickly adopt those are very general ability skills and – or understanding. Generally these systems are sort of messaging based so having some knowledge or done some research on how messaging based systems work*

Innovation capacity (+): Supported – In the context of the MLO innovation is hampered by the lack of integration. However, an EDA has a capacity for innovation.

“The system definitely is something that scales very nicely for new features. There’s no limitations to how much you can add to the system...which is good...” The decoupling of the services provides for agility in adding new features: *“Yeah I think it kind of it’s event driven architecture but it’s also it also relates to your system services. So when you have these decoupled systems and you have the ability to throw in any service and listen to events. It will naturally make your system more agile.”* Adding new features and integrating with third party components it is much simpler in performing this through an event driven approach: *“Event driven architecture tends to make using third parties far simpler and easier.”*

Knowledge capability (+): Partially supported – In the context of the MLO obtaining knowledge is a challenge due to difficulty obtaining information. Software engineers that are newcomers to the distributed architecture and in particular to an EDA solution

have to quickly adapt and learn. The MLO has a number of resources available for software engineers to learn the skills required to develop and maintain the EDA solution. *“Distributed systems experience generally... I think it’s quite poor and the candidates I see and so generally I’m trying to look for people who have more the capability as opposed to the existing knowledge and you know there’s a lot more on the job learning.”* A main concentration point is the MLO wiki that contains information from all the regions on specific details on the system. This is mandatory requirement in introducing software engineers to knowledge of distributed systems. *“At the moment just feel we relying on so many little bits and pieces to help us with our one core objective. It’s very fragmented put it that way.”*

Operational capability (+): Supported – A generalisation of this capability is that the decoupling of an EDA helps in achieving an operational capability by modularisation. The monolithic system prior to the adoption of the EDA was difficult to maintain and develop new features. This is stated by the SDM contrasting both systems: *“More monolithic architecture which was based a lot on store procedures and triggers to cause work to occur and because of that it was an extremely difficult system to maintain and to develop on and everything was one big code blob and it you know created lots of problems in terms of having lots of people work on it.”* This is in contrast to the EDA solution: *“...so moving to the EDA solution was a huge improvement in terms of you know having things broken down into smaller pieces and therefore you know little bit more understanding”*

Top management support (+): Supported - Top management refers to management that make strategic decisions and commits resources to the completion of projects to obtain a strategic goal. Support from management in adopting an EDA is paramount. Although the management team understand there are complexities, very few understand the extent of those complexities. *“They don’t all understand the complexities I think they all understand that there are levels of complexities... Uh....varies from person to person some more technical than others and some just really shouldn’t be worrying about some areas of technicality.”* Management has concerns over certain areas. The main issue is not with the EDA solution but the process and resources in accomplishing a change to the system. *“Management is constantly frustrated by the length of time it takes to make changes. I think that one of the biggest challenges there is the lack of continuity...”*

5.3 Environmental Context

Legislation factors (+/-): Partially supported – Micro lending in the digital marketplace is a relatively new industry. Legislation and regulations play an important part on shaping the micro lending industry. The MLO due to its geographical dispersion has to contend with multiple jurisdictions. Legislation factors do *not* have a direct influence on the adoption: *“I don’t think regulation in and of itself is going to be making a difference to whether one uses an event driven architecture or not”*. However, an EDA helps in achieving flexibility as in the context of the MLO, so from a regulatory perspective there is an advantage to having a service-oriented approach as opposed to a monolithic approach: *“...so from so from a regulatory stand point – I guess there is an advantage to service-orientedness as opposed to monolithic publicationess...”* since *“agility [is]*

the ability to uh...react timeously [to legislative changes] without having to touch every part of the system.”

Competitive pressures (+/-): Partially supported – an organisations ability to adapt to competitive pressures is reliant on the underlying information systems. Although the architecture itself is not a competitive advantage, it does provide the capabilities to deliver features and create new products. However, these information systems can be implemented with different architectures. *“...so it’s not the architecture itself that I think would give the competitor an advantage it’s how you use that architecture.”* A competitive pressure that is experienced by the MLO is competitors that utilise the service to determine how to better the MLO’s customer experience. *“There’s naturally the ability for your competitors to swoop in and kind of look at your journey and where things fall flat and be able to add that value to the customer...”* Therefore, the competitive pressure of continuously adapting and being agile to prevent customer loss is reliant upon the underlying applications that in turn are reliant on the architecture.

Customer mandate (+): Unsupported – factors do not influence the adoption. However the capabilities offered, provide an advantage. Information and the interaction of customers provide a means of driving innovation: *“Our business is enabled through technology. In other words, we got a customer that goes onto a website and is able to do a bunch of things in order to interact with us.”* A more general discussion not related to the adoption of an EDA but to customers of the MLO is that customers form part of the core of the MLO and if these customers are not adequately serviced, they will move to a competitor: *“...I think there’s three core effects by not delivering to the customer. Essentially I mean that there’s a reputation ...there’s naturally the ability for your competitors to swoop in and kind of look at your journey and where things fall flat third level there’s obviously a sustainability the more customers you start losing and you know that affects the reputation and the customers defecting to competitors.”* Thus this does not necessarily influence the adoption of an EDA but the capabilities offered by an EDA provide an advantage.

5.4 Correlations between Factors

During the analysis, we found that there are many of these factors influencing the adoption of EDA correlate with each other. The analysis showed a correlation between factors such as complexity and cost, innovation and human resources and vendors, tools and operational capability. By taking advantage of EDA capabilities, distributed systems become reliant and agile in accomplishing the objectives of an organisation. Future research could investigate these correlations in more detail.

6 Limitations

The research conducted a case study within a single organisation. As such, the research provides a single example of the adoption of an EDA. Since very little research has been carried out on the adoption of EDA, correlating the results with other research is difficult. The micro lending case study is carried out within the financial sector and thus there is little knowledge of how the adoption of an EDA relates with other domains

although if the capabilities offered by an EDA provide an advantage this could be translated to other domains.

7 Conclusion

EDA provides an organisation's enterprise systems with many capabilities, including the publish-subscribe pattern, asynchronous communication and decoupling. These enable organisations to make real time decisions in dispersed information systems. Decoupling achieved by employing an EDA improves scalability allowing for large-scale distributed environments. Capabilities offered by EDA allow organisations to manage business processes more flexibly by allowing real time access. EDA enables IT architecture in organisations to respond and take action across the organisation levels. This is accomplished by allowing interoperability across information systems and as a result enables flexibility in the organisation. Micro lenders of short-term loans are well suited to the adoption of an EDA. These micro lenders require an information system that can process events in real time. This real time processing must provide verifications and decisions on customers taking out loans.

The analysis of the case study affirms the capabilities that were highlighted in the literature review. These capabilities play an important part in enabling the MLO to be adaptable, flexible and reliable. The factor of relative advantage highlighted that the technical abilities of an EDA namely scalability, asynchronous communication and decoupling are pre-requisites for realising these capabilities. Further to this, an advantage over SOA is that EDA limits temporal coupling. The benefits of the capabilities allow the MLO and other organisations, adopting an EDA, to have a reliable and fault tolerant system that can help in achieving its objectives.

However, these capabilities do come at a cost. The cost is associated not only to financial implications of a distributed system but also introduces more complexity. This complexity is associated with the increases in infrastructure, resources and tools. Furthermore, the skills and experience levels of staff is of great importance. From a software engineering perspective, skills in asynchronous communication is paramount to the understanding of the system. These personnel are drivers of innovation. A concern raised is that the lack of experienced technical resources hampers innovation. In the context of the MLO, the generation of information from the EDA to obtain knowledge, is complicated by the complexity of the data structure of the system. Managing the structure requires greater skills and resources. Hence this research advocates for more distributed system knowledge to be introduced at tertiary level and more is needed to educate entry-level software engineers about distributed architectures.

The fit of the technology namely EDA to software engineers and business personnel highlighted a few possible areas of improvement of the MLO. A recurring concern is around information extraction and the centralisation of information. This has added implications in the relationship, quality and compatibility of tasks and processes to the EDA solution.

It is difficult to extrapolate to what extent our findings are generalizable to other organisations. The case study description given should allow one to draw parallels between organisational contexts and, to the extent that contexts are similar, we would think that at least some of the factors will apply as well. Generalising across a wider

sample of organisations, especially in the service industry, would be a recommended avenue for future quantitative research.

This research should of interest to Information Systems researchers as well as practitioners and businesses looking to see a real-world application of EDA as well as the important adoption issues and factors that were experienced in the case study.

8 References

1. Chen, Y., Wang, Y., Nevo, S., Jin, J., Wang, L. & Chow, W.S. 2013. IT capability and organizational performance: the roles of business process agility and environmental factors. *European Journal of Information Systems*. 23(3):326–342.
2. Magoutas, B., Riemer, D., Apostolou, D., Ma, J., Mentzas, G. & Stojanovic, N. 2013. An event-driven system for business awareness management in the logistics domain. In *Business Process Management Workshops*. 132: 402–413.
3. Juric, M.B. 2010. WSDL and BPEL extensions for Event Driven Architecture. *Information and Software Technology*. 52(10):1023–1043.
4. Clark, T. & Barn, B.S. 2012. A Common Basis for Modelling Service-Oriented and Event-Driven Architecture. In *Proceedings of the Fifth India Software Engineering Conference*. 23–32.
5. Krumeich, J., Weis, B., Werth, D. & Loos, P. 2014. Event-Driven Business Process Management : Where are we now ? – A Comprehensive Synthesis and Analysis of Literature. *Business Process Management Journal*. 20(4):615–633.
6. Vezzani, R. & Cucchiara, R. 2010. Event driven software architecture for multi-camera and distributed surveillance research systems. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*. 1–8.
7. Raffelsieper, T., Becker, J., Matzner, M. & Janiesch, C. 2012. Requirements for a Pattern Language for Event-driven Business Activity Monitoring. In *European Research Center for Information Systems (ERCIS)*. 1–12.
8. Ave, W.M., Kung, L. & Byrd, T.A. 2013. Leveraging Event-driven IT Architecture Capability for Competitive Advantage in Healthcare Industry : a Mediated Model. In *Proceeding of Thirty Fourth International Conference on Information Systems*. 1–12.
9. Luckham, D. 2011. *Event processing for business: organizing the real-time enterprise*. New York, NY: John Wiley & Sons.
10. Nitz, S., Kleiner, C., Koschel, A. & Astrova, I. 2013. Applying event-driven architecture to mobile computing. In *IEEE International Symposium on Signal Processing and Information Technology*. 58–63.
11. Fan, M., Fan, H., Chen, N., Chen, Z. & Du, W. 2013. Active on-demand service method based on event-driven architecture for geospatial data retrieval. *Computers and Geosciences*. 56(1):1–11.
12. Esposito, C., Cotroneo, D. & Russo, S. 2013. On reliability in publish/subscribe services. *Computer Networks*. 57(5):1318–1343.
13. Barthe-Delanoë, A.-M., Truptil, S., Bénaben, F. & Pingaud, H. 2014. Event-driven agility of interoperability during the Run-time of collaborative processes. *Decision Support Systems*. 59(1):171–179.
14. Zagarese, Q., Canfora, G., Zimeo, E., Alshabani, I., Pellegrino, L., Alshabani, A. & Baude, F. 2014. Improving data-intensive EDA performance with annotation-driven laziness. *Science of Computer Programming*. 97(1):266–279.

15. MacLennan, E. & Van Belle, J.P. 2014. Factors affecting the organizational adoption of service-oriented architecture (SOA). *Information Systems and e-Business Management*. 12(1):71–100.
16. Basias, N., Themistocleous, M. & Morabito, V. 2013. SOA adoption in e-banking. *Journal of Enterprise Information Management*. 26(6):719–739.
17. Ghalsasi, S.Y. 2009. Critical success factors for event driven service oriented architecture. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*. 1441–1446. ACM.
18. Clark, T. & Barn, B.S. 2011. Event driven architecture modelling and simulation. In *IEEE 6th International Symposium on Service Oriented System (SOSE)*. 43–54.
19. Mashigo, P. 2012. The lending practices of township micro-lenders and their impact on the low-income households in South Africa: a case study for Mamelodi township. *New Contree*. 65(1):23–46.
20. Morse, A. 2010. Payday lenders: Heroes or villains? *Journal of Financial Economics*. 102(1):28–44.
21. Nkpoyen, F., Bassey & Eteng, G. 2012. Micro - Lending As an Empowerment Strategy for Poverty Alleviation among Women in Yala Local Government Area of Cross River State, Nigeria. *International Journal of Business and Social Science*. 3(18):1–9.
22. Bhutta, N. 2014. Payday Loans and Consumer Financial Health. *Journal of Banking & Finance*. 47(1):230–242.
23. Gangwar, H., Date, H. & Raoot, a. D. 2014. Review on IT adoption: insights from recent technologies. *Journal of Enterprise Information Management*. 27(4):488–502.
24. Tornatzky, L. & Fleischer, M. 1990. *The process of technology innovation*. Lexington, MA: Lexington Books.
25. Saunders, M., Lewis, P. & Thornhill, A. 2009. *Research Methods for Business Students*. 5th ed. Harlow: Prentice Hall.
26. Betts, D., Domínguez, J., Melnik, G., Simonazzi, F. & Subramanian, M. 2012. *Exploring CQRS and Event Sourcing*. Redmond, WA: Microsoft.
27. Boike, D. 2015. *Learning NServiceBus*. Mumbai: Packt Publishing Ltd.