

Configuration information system architecture: Insights from applied action design research

Asif Qumer Gill*, Eng Chew

University of Technology Sydney, Ultimo, NSW, 2007, Australia



ARTICLE INFO

Keywords:

Action design research
Architecture design
Configuration information system
Resilience
Service

ABSTRACT

One of the critical information systems that enables service resilience is the service configuration information system (CiS). The fundamental challenge for organisations is the effective designing and implementation of the CiS architecture. This paper addresses this important research problem and reports insights from a completed applied action design research (ADR) project in an Australian financial services organisation. This paper aims to provide guidance to researchers and practitioners contemplating ADR, rooted in the organisational context, for practice-oriented academia-industry collaborative research. This research also contributes in terms of the CiS reference architecture design knowledge and demonstrates the applicability of the ADR method.

1. Introduction

Financial services organisations (FSOs) offer a range of services to their customers such as insurance, wealth, payment and advisory services [1,2]. Service resilience is the ability to effectively defend, respond and evolve in response to expected or unexpected changes or disruptions [3,4]. These disruptions could include service failure, datacentre facility failure, equipment obsolescence, unforeseen inter-system interactions, human errors, cyber-attacks, regulatory changes, market changes, customer changes and technology changes. Ineffective service resilience may cause frequent disruptions, thus resulting in a poor customer satisfaction and regulatory implications or financial losses [5,6].

A service can use other services and depends on a number of configuration items (CIs) or components (e.g. people, processes, information, applications, platforms, infrastructure and facility). Thus, complex dependencies and integration management of several internal and external services of an organisation are a reality. A service configuration information system (CiS) is one of the critical systems that can help in enabling service resilience. The CiS can be designed for managing the information about the complex dynamic relationships hierarchy of services and its underlying components. The information from the CiS can be used for effective impact and root cause analysis of a disruptive event and eventually defend, respond and evolve to ensure service resilience. The fundamental challenge for organisations is the effective designing and implementation of the CiS architecture [7]. The strategic importance of the CiS design (a.k.a. CMDB), as a particular class of

information systems, has long been recognised [8,9].

It has been found that often CiS-type projects, under the guise of consultant and vendor specific configuration management databases (CMDBs), have failed in the past. It has been reported that ‘many organisations each year who attempt to build a CMDB have the best of intentions—but the overwhelming majority, an estimated 85 percent, fail in their efforts’ [10]. Lack of active stakeholder engagement (e.g. developers, providers and users of the CiS) and bottom-up technology-centric solutions are some of the important reasons for such failures [11]. Traditional CiS solutions focus on a big bang waterfall method of delivery (as opposed to an incremental delivery) by developing and evaluating the CiS in separate serial stages with little to no engagement of users during the development stage (e.g. lack of social design and value co-creation).

The FSO (our industry partner) engaged UTX (University) researchers in 2014 to research and define an alternative approach to effectively design and implement the CiS architecture, which is the main focus of this paper. The FSO mentioned that ‘we approached this project aware that there was a history of large, expensive, complex projects to deliver corporate configuration management databases with an unfortunately high failure rate. Thus we determined that a different approach was required if we were to improve our chances of success. Our CIO reached out to researchers at “UTX” to utilise their experience and frameworks to provide a unique alternative to “break the mould.”’ Further, in the initial project brief from the FSO senior manager operations, it was highlighted that the traditional methods to the CiS have failed, and we must find an alternative approach – one that is consistent with the FSO

* Corresponding author.

E-mail address: asif.gill@uts.edu.au (A.Q. Gill).

approach to information systems development. Thus, the FSO engaged UTX to address the following important practice-oriented research question in hand.

RQ: How to effectively design and implement a service configuration information system (CiS) architecture?

We address this important practice-oriented research problem by applying the action design research (ADR) method [12] for effectively designing and implementing the CiS architecture in an Australian FSO. Thus, this paper reports insights about the applied CiS architecture design and implementation approach from our completed ADR project (2014–2016) in the FSO, Australia. This project began in April 2014 at the FSO, Sydney, Australia, and continued until August 2016. The scope of this paper is limited to the CiS and ADR approach. There is only a scarce literature on both the CiS design and the application of ADR in real industry projects. This paper is an attempt to fill this small research gap. This paper can be used by both researchers and practitioners contemplating ADR for effectively delivering practice-oriented collaborative architecture design research projects similar to the CiS.

This paper is organised as follows. First, it provides the research background and problem in the context of the FSO. Second, it discusses the adaptive ADR method, which has been applied to address the identified problem. Third, it presents insights from the application of ADR research project to the FSO in Sydney, Australia (2014–16). Finally, it discusses the research results followed by conclusion.

2. Research background: motivation and problem

The term resilience comes from the Latin word *resiliere*, which means bounce back [13]. The concept of resilience can be traced back to the field of ecology [14], reliability engineering and system safety [15]. The concept of resilience has evolved in the mid-1990s and early 2000 in the multi-disciplinary contexts of human natural systems [16]. Academic and practitioners' community is showing great interest in resilience. A new body of work and research interests are evolving in this space focusing on the resilience of complex adaptive socio-ecological systems [17]. Despite this increasing interest, there is a lack of research on organisational resilience, in particular service resilience [21,18,8].

Resilience is the ability of a system to absorb, adapt and recover rapidly from a disruption so that normal levels of the service delivery can resume [19]. Resilience can be characterised by a number of properties such as defence or resistance to attacks, efficiency or recovery (immediate response for stability) and adaptability or evolution (to a new 'better' state) [18]. These capabilities can be supported by a number of resilience processes: anticipation, monitoring, responding and learning/adapting [20]. Resilient organisations or high reliability organisations (HROs) or learning organisations [21] demonstrate their mastery by balancing systemic stability and efficiency with flexibility and adaptability, which allows for adaptations in the face of uncertainties and disruptions without losing control [19].

Financial service enterprises offer a range of financial services to their customers [22]. Their services could be impacted by a number of expected or unexpected disruptions [1]. The concept of service resilience, in the overall context of organisational resilience, is receiving significant interest among financial services [22]. Financial services need to offer and run resilient services to operate effectively under stress and to have the ability to adapt in response to disruptions to avoid any potential losses [23]. A resilient service would be able to quickly recover from a disruption or disaster state and not experience undesirable long-term service disruptions [24].

Service resilience is a complex organisational phenomenon and involves a number of systems, users, operations and design communities. One such system is the CiS. The CiS can be linked to an existing research area of digital infrastructures [53]. Digital infrastructure is described as a 'shared, unbounded, heterogeneous, open, and evolving socio-technical systems comprising an installed base of diverse information

technology capabilities and their user, operations, and design communities' [54]. The CiS, as a digital information infrastructure for services and their components, is dynamic, relational and recursive in nature. For instance, organisations and their services (e.g. number, type, version and configuration of services) evolve to meet the dynamic needs of their customers. This indicates the need for a dynamic and scalable digital information infrastructure for managing dynamic service information. Further, the CiS needs to allow the users to map the relationship between services and their components and allow the services to be recursively organised, packaged to create new services or applications and discovered for impact and root cause analysis. This clearly links the CiS to the fundamental characteristics of the digital infrastructure such as flexibility, recursive nature, reconfiguration of services and creation of applications or services by aggregating, integrating or making use of other services and low-level components [55,56]. The digital infrastructure provides useful lenses for conceptualising the CiS as an integrated information system of systems for managing services and their information [57].

The FSO senior management intends to develop a CiS, which can provide the critical information about services (digitisation of services) and its underlying components for performing the impact and root cause analysis of disruptive events or changes for ensuring the service operations and resilience. However, the problem is the effective designing and implementation of the CiS. As discussed earlier, existing waterfall and technology-centric solution delivery approaches for designing and implementing the CiS architecture as a CMDB have failed in the past [10]. This research paper aims to address this important research question by using the ADR method for designing and implementing the CiS architecture appropriate to the FSO context. The impact of the CiS on FSO and its resilience is a separate research project, which is not within the scope of this paper.

The ADR research begins with motivation and a practical problem. In this ADR project, the CiS architecture design and implementation is the motivation and a real practical problem for the FSO. The next step is the proposed CiS solution, its iterative development and evaluation. Hence, the motivation, problem and proposed solutions are clear. However, as discussed earlier, the challenge is the designing and implementation of the CiS architecture. This challenge is addressed by a consultant and vendor-independent practice-oriented research-based approach of ADR as appropriate to the FSO organisational context (e.g. innovative mindset, people and culture, policy and principles, executive sponsorship and highly motivational practitioners who are prepared to engage in this practice-oriented research and improvise).

3. The ADR method for the FSO

There are a number of quantitative and qualitative approaches to address the Information Systems (IS) research. The quantitative approaches are mainly data driven and apply various statistical analysis methods, whereas the qualitative approaches are mainly case study driven and apply interpretive or constructive methods. In the last two decades or so, interventionist approaches of practice research, such as action and design research, are gaining popularity and are considered appropriate for IS research. Action research (AR) and design research (DR) are the two practice-oriented information systems (IS) research methods that aim to address a practical phenomenon in its practical context. These methods are gaining considerable attention from the IS community, which is evident from the recent studies published in high-quality IS journals [e.g. 12, 25, 26].

AR is an interventionist research method, which aims to solve a practical problem through intervention and collaboration between researchers and practitioners [27]. DR is a constructive research method, which aims to solve a practical problem through building and evaluating the artefacts [28]. The main difference between the AR and DR is the artefact. Artefact and knowledge about the artefact are considered as knowledge contribution in DR. Unlike DR, AR is focused on the

improvement of practice and knowledge body through intervention.

The history of DR can be traced back to the 1960s [29]. The earlier work was mainly focused on building architecture and industrial design, which was further extended to engineering design in the 1980s. This work resulted in the establishment of DR as a new discipline. The discipline continues to grow as a practice research method [28,30–3238]. DR, sometimes, is referred as design science research (DSR). However, the theoretical nature of ‘science’ and practical solution orientation of ‘research’ set these concepts apart [33]. The other school of thoughts and terms have been appeared over time such as ‘sciences of design,’ ‘design as a research,’ ‘research as a design’ or ‘practice as design.’ However, the current work refers to DR as a constructive approach to designing and evaluating artefacts (e.g. construct, model, method, capability and framework) as a solution to a research problem in hand [28]. Being a practice research method, DR always attracts criticism on methodological rigor, which is required for academic contribution [34]. Further, it has been argued that DR artefact should be emerged from the organisational context to deal with a class of problems [35]. Existing DR methods need to include organisational context elements [36]. Further, it has been reported that ‘existing DR methods focus on building the artefact and relegate evaluation to a subsequent and separate phase’ [25], which is sometimes not feasible where iterative building, intervention and evaluation of the artefact are required through continuous interactions with organisational context elements. Hence, the problem with traditional DR is twofold: (1) the artefact needs to be emerged through iterative interactions with organisational context elements and (2) artefact building, intervention and evaluation need to occur iteratively. To address this concern, we adopted an integrated ADR approach for the FSO, which integrates the best of DR (iterative artefact development and evaluation) and AR (intervention and organisational context) elements [37].

Thus, this research project applied the ADR method for solving the practical design problem of the CiS architecture through the intervention with the aim of iteratively developing and evaluating the CiS solution artefacts in the practical context of the FSO. The ADR is a consultant and vendor-independent approach to address the practical phenomenon within the practical context of FSO – a practice research [39]. The ADR aims to release research results early and frequently in short iterations as the project progresses. The ADR method incorporates three main elements: adaptive, action and design research (see Table 1).

The ‘adaptive’ element in the ADR reflects the iterative [40] nature of the ADR method. As opposed to the big bang delivery of ADR project artefacts (such as the CiS) in top-down and separate development and evaluation stages, this research iteratively developed (designed) and evaluated (implemented) the CiS architecture. This allows the ADR project stakeholders to see early results, provide continuous feedback and make any necessary adjustments both in the method and artefacts for continuous adaptation.

The ‘action’ element reflects the interventionist and collaborative natures of the ADR method, which is typically aimed to contribute to the knowledge body by solving a practical problem [27]. Action research occurs in a natural environment of the host organisation where researchers and practitioners actively engage and collaborate to solve a practical problem through intervention [41]. Action research is rooted

in positivists and post-positivist constructive ideas [27]. Action research can be done in a number of stages, some of the most common stages are as follows [42]:

- Identification of the problem
- Planning action research and execution of action research steps
- Evaluation
- Learning

The ‘design’ element reflects the constructive nature of the ADR method, which typically aims to contribute to the knowledge body by developing and evaluating an artefact such as construct, model, method (e.g. algorithm, practice, capability and framework) and instantiation (e.g. software and implementation). Design research is a constructive approach for designing an artefact as a solution to a practical research problem in hand [38]. Similar to action researchers, design researchers collaborate with practitioners in developing and evaluating the artefacts for solving a practical problem. However, it has been reported that design research is more conceptual in nature and not grounded in practice [12]. However, design can be generalised (e.g. reference design or architecture) and transferable to different contexts. Further, design research is a staged approach and focuses on the designing and evaluation of artefacts in separate serial stages. Peffers et al. [43] provide the following stages:

- Identification of the design problem and objectives
- Designing and development of the artefact
- Demonstration and evaluation
- Communication

This research applied an integrated (AR and DR) research method known as ADR to address this practical research problem for the FSO. The ADR method seems appropriate to address the research problem in hand that requires both intervention and iterative CiS artefact development and evaluation. The ADR (see Fig. 1) has been adopted on the basis of the existing ADR concepts (see Table 3) [12,42], stages, principles and practices or tasks [25]. The applied ADR ‘practice’ in the FSO, and the resultant CiS artefacts and knowledge about the artefact ‘design’ are the key outcomes, which can be generalised, modified and transferred to other organisational contexts. A conscious effort was made to align the ADR to the FSO context, thus resulting in an iterative project delivery approach that defines a value co-creation process involving research and practice. Thus, the ADR method is organised into four stages (idea for research, problem formulation and objectives, BIE and RL, and formalisation of learning and final feedback) (based on [25]), in alignment with the FSO context and project delivery approach (Initiative, Initiate, Discover, DevOps and Close), to iteratively design and evaluate the CiS (see Fig. 1):

- 1 Initiative (intuition/ idea) [idea for research]
- 2 Initiate [problem formulation and objectives]
- 3 Discover and DevOps [Iterative building, intervention and evaluation (BIE) and reflection and learning (RL)] and
- 4 Close [formalisation of learning and final feedback].

Table 1
ADR Method - based on Papas et al. [12].

Items	Epistemology	Axiology	Ontology	Theory Generation	Methods
Action Research	Constructivist Interpretive, positivist and post-positivist epistemology	Learning and improved practice	Intervention and collaborative	Contextual, repeatable with some generalisability	Qualitative Grounded theory can be used.
Design Research	Pragmatism Interpretive and post-positivist epistemology	Artefact development and evaluation	Conceptual artefact development and evaluation	Generalisability and transferability of artefact design	Qualitative Development and evaluation of artefacts (e.g. models and software) in separate stages

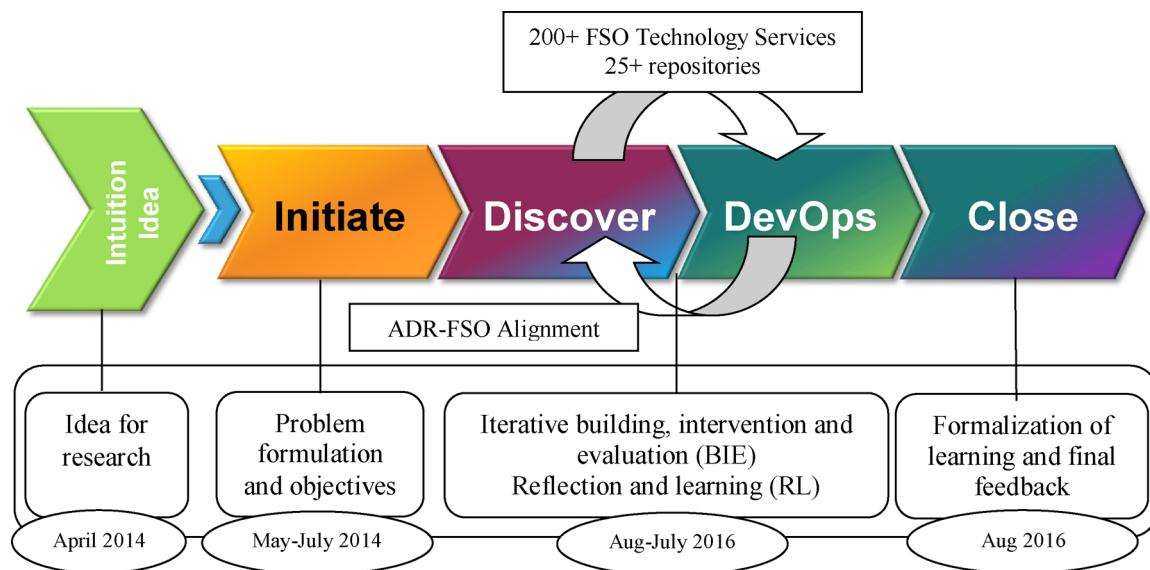


Fig. 1. ADR and FSO alignment (value co-creation process).

Here, a key insight is that researchers' methods or ways of working need to be aligned to the industry research partners' problem and ways of working. Nevertheless, the focus of such applied research is to address the problem of stakeholders. Thus, this project is a good mix of theoretical rigor and practical problem or reality. Thus, the ADR echoes not only the theoretical precursors and intents of the researchers but also the influence of the involved stakeholders or users for iterative value co-creation in the social and practical FSO contexts. The ADR focuses on the iterative co-creation and evaluation of the desired artefacts (see the section on ADR in Practice). It does not follow the traditional stage-gate model of DR in which building and evaluation are done in a linear order or sequence. The ADR method contains iterative principles, iterative stages and practices that address these issues. The ADR for a single FSO case could be considered a limitation when generalising the results of this study. As discussed earlier, the literature on the application of ADR in real industry projects is scarce. This paper demonstrates a concrete application of the ADR in a more than 2 year-long industry project. Further, it lays a foundation for future ADR cases in this important area of architecture-driven CiS design and implementation for service resilience. Thus, despite the single case limitation, this paper contributes, through a real-life practice, new knowledge on the designing and implementation of the CiS architecture using the ADR. Further, this 2 year-long project clearly demonstrated the value contribution and appropriation of ADR in solving a practical and complex problem of FSO.

4. Application of ADR in practice

A controlled linear DR artefact building and evaluation (BE) approach is often not feasible to address the complex dynamic problems where a more adaptive approach is required to iteratively discover the solution in small increments. Thus, an ADR has been applied to address the 'CiS as a class of a problem' encountered in the specific organisational setting of the FSO through iterative building, intervention and evaluation (BIE) of the CiS architecture artefacts. For instance, there are more than 200 FSO technology services and more than 25 repositories (discovered or identified during this project) (see Fig. 1). As opposed to a big bang traditional approach to designing and implementing the CiS for all services and integrating all the repositories for a single aggregated view of service and underlying components information, an

initial list of six high-value services and three related service component data repositories (service changes/incidents/requests management, physical and virtual infrastructure components) was identified to design the initial CiS architecture. Additional services and repositories were discovered and included as the project was iterated. This architecture was continuously evaluated and evolved to accommodate other services (subject to service resilience) and repositories iteratively during the BIE and RL stages (see Fig. 1). This section provides insights about the ADR stages, each anchored by the principles and practices (Table 3) applied to the designing and implementation of the CiS architecture at the FSO. This project involved researchers from UTX (coded name) who actively worked (action research intervention) with the FSO (coded name) employees (practitioners including external vendors) to design the CiS artefacts (design research) for addressing the problem of service resilience.

4.1. CiS as a class of problem

Before reflecting on how the ADR was applied to the CiS project, we identify and describe the class of problems in focus: service configuration information management in service organisations. The FSO offers a range of business services to their customers [1]. The FSO operates in a highly competitive environment that requires the support of technology services for maintaining the quality, brand and consistency of business services while continuously defending, responding and evolving to meet the complex and dynamic needs of their customers [2]. Service configuration information management is a complex undertaking for the FSO operating in the always changing business, technology and regulatory environment. Ad-hoc and inept service configuration information management may result in periodic disruptions to operations.

Service configuration information management involves the monitoring and analysis of the large amount of operational service data and involves a number of processes associated with the services and underlying service components (technology and non-technology components), which are broadly known as service CIs. In this context, a service, being a part of another service, is also a CI.

As discussed earlier, organisations have long identified the importance of the CiS in managing service configuration information [7,8]. As a particular class of information systems, the CiS can be

Table 2
Kernel Theories.

#	Item	Description
1	Adaptive enterprise service system (AEss) [46]	To design the CiS, we needed the reference meta-model. Thus, we used the AESS, which provided the service-focused taxonomy to design the CiS. The purpose of the CiS was to capture the FSO services information for reporting, dependency analysis, root cause analysis, etc. to manage service resilience. This has been further explained in detail in the paper (e.g. see Fig. 5).
2	Design thinking (DT) [45]	Design thinking offers a balanced approach or mindset of intuition and analytical thinking, which was used for the continuous design or re-design of the CiS in small iterations on the basis of the feedback loop mechanism. Design thinking is clearly evident in the four stages of applied ADR.
3	Living systems theory (LST) [47]	Living systems theory was used as lenses to conceptualise the inter-connected services and their components for the CiS design. The systems thinking was important to capture the requirement that a change in one service or component may have an impact on other services or components in the CiS.
4	Service science (SS) [48]	Service science views an individual or organisation as a ‘service system.’ This was used to conceptualise the FSO as a service system, where the fundamental unit is ‘service’ as opposed to low-level resources or assets such as servers and devices. This provided a more service centric view of the FSO.
5	Model-driven architecture approach (MDA) [49]	Model-driven architecture provides three layers of architecture abstraction: conceptual, logical and physical architecture layers. The CiS design was organised and explained in terms of these layers.

specifically designed to help organisations in managing information about services, underlying components and their complex hierarchical and ontological relationships. The CiS can facilitate searching information for and mapping specific services, components and their dependencies for root cause, impact analysis forecast and trends analysis. The challenge is that CiS-type projects have been failed in the past (as discussed earlier), and this served as a stimulus to initiate this research project to address this challenge by effectively designing and implementing the CiS architecture at the FSO. One of the reasons of such failures could be attributed to the consultant and vendor-specific solutions, bottom-up technology asset management focused (e.g. applications, infrastructure and device) approach to CiS implementation as opposed to a top-down service-centric and model-driven architecture approach. This was a challenge for the CiS project team at the FSO. This led to the engagement of researchers from UTX by the FSO in Sydney, Australia. Several other external vendor organisations (e.g. Assyst and Ratify) also participated from time to time to support this project in the review and development of the CiS architecture.

4.2. CiS project

This project began in April 2014 at the FSO, Sydney, Australia, and continued until August 2016. Researchers from UTX were approached by the FSO in 2014 to help in the designing and implementation of the CiS architecture for service resilience.

4.2.1. Team

This project team involved two researchers from UTX who actively worked (ADR intervention) with the FSO team to design the CiS artefacts for solving the problem at hand. The core of the FSO team members (10+) comes from enterprise strategy and architecture (enterprise architect), project delivery (business analyst, project manager and development team) and operations (operations delivery manager, analysts and senior manager) areas. It is also important to note here that users (e.g. incident manager, infrastructure manager and admin staff) are also a part of the team and come mainly from operation areas. This is because the CiS scope is limited to technology operational resilience; thus, users from technology operation areas were involved in the co-creation and evaluation of the CiS via feedback loops or sessions. External vendor organisations were also engaged by the FSO to support the CiS project at different stages in particular external architecture design review (evaluation of the artefacts through external experts), solution development (coding) and user experience design activities (e.g. CiS style guide and user experience journey mapping). Unlike traditional approach to CiS delivery, it is important to note here that users of the CiS were identified at the initial stage of the project (as

opposed to that only at the final stage of big bang evaluation or testing) and were part of the core team and involved throughout the project for actively evaluating the artefacts and providing the feedback to support the iterative BIE and RL activities. This indicates the connotation to ‘emergence’ property of ADR – this is the reason why it is suitable to the FSO context as the problem space is uncertain (e.g. users are unsure of their needs) that needs ‘just do it’ and ‘see what happens’ then ‘reflect and readjust’ kind of design/implement process.

4.2.2. Goal

The goal was to discover and test design principles and practices to address the identified research problem of the CiS design and implementation. It was anticipated that these principles and practices would provide an important theoretical and practical contribution that would assist in the design and implementation of CiS-type projects at the FSO and similar organisations.

4.2.3. Kernel theory

Kernel theories can be used to influence or inform the design research [44]. Kernel theories (e.g. reference architectures and metamodels) can provide baseline generic elements for designing context-specific ADR artefacts. The kernel theories used were adaptive enterprise service system (AEss) [46], design thinking (DT) [45], living systems theory (LST) [47], service science (SS) [48] and model-driven architecture approach (MDA) [49]. The action-design research team combining researchers and industry professionals applied the AESS framework [46] as a reference architecture or meta-framework (kernel theory) in this research project. The AESS was used because it provides a vendor-independent, research-based service metamodel, which was used to inform the development of the service-focused CiS architecture. Thus, the AESS metamodel and concepts from DT, LST, SS and MDA with relevant industry best practices were used to guide the design and implementation of the CiS architecture for resilient services in the Australian FSO local context (Table 2).

4.2.4. Existing FSO artefacts

In addition to kernel theories, existing FSO repositories, models, platforms, systems and relevant knowledge about the inner FSO operating environment or context (e.g. strategy, architecture principles, policy, asset management repositories, operations architecture and process models) were identified and used. These existing artefacts supported the formulation of design concepts for developing the models for the CiS architecture. This clearly indicates that the ADR-generated artefacts need to adhere to the FSO context.

Table 3

Insights from the applied ADR: stages, principles, practices and artefacts.

Stages	Principles	Practices	Artefacts
Stage 1: Initiative (intuition or idea) (April 2014)	Principle 1: Practice-Inspired Research	Project initiated by the FSO (practice driven) Ideation through initial engagement and investigation Non-disclosure agreement (NDA) development	Project idea document Non-disclosure agreement (NDA)
Stage 2: Initiate (Problem Formulation: vision and scope) (May 2014 – July 2014)	Principle 1: Practice-Inspired Research	Strategy review and project vision and scope definition Project planning and funding Contract and funding for project (Discovery and DevOps) – 1st payment Work streams definition	Project vision and scope Architecture principles Project plan Project contract Work stream structure Kernel theories
	Principle 2: Theory-Ingrained Artefact	The following key kernel theories were identified and reviewed to inform the practice-oriented research: Adaptive enterprise service system (AESS) Design thinking (DT) Living systems theory (LST) Service science (SS) Model-driven architecture approach (MDA).	
Stage 3: Discovery and DevOps (August 2014 – July 2016) <i>Stage 3a: Discovery and DevOps (Building, Intervention and Evaluation)</i>	Principle 2: Theory Ingrained Artefact Principle 3: Reciprocal Shaping Principle 4: Mutually Influential Roles Principle 5: Authentic and Concurrent Evaluation Principle 6: Guided Emergence	Iterative detailed research and analysis Knowledge, solution streams and user collaboration Iterative artefact development, evaluation and showcase for services subject to resilience Communication, adjustment and publications	CiS architecture context model Conceptual architecture model Logical architecture model Physical architecture model (Implementation) Proof of concept Architecture review
<i>Stage 3b: Discovery and DevOps (Reflection and Learning)</i>			CiS architecture showcases and demonstration for feedback Updated CiS architecture and solution elements New emerged CiS architecture and solution elements 2 Work-in-progress academic papers Design: CiS design principles; architecture reference models Practice: ADR principles and practice emerged as an innovation process
Stage 4: Close (Formalisation of Learning) (August 2016)	Principle 7: Generalised Outcomes	Project retrospective, final review, handover and closure	

4.2.5. Existing industry principle and practices

The CiS is an information-intensive system. Therefore, the existing information principles (e.g. information has a common definition and information is managed as an asset) of FSO were analysed and used to guide the CiS architecture. The CiS project was mainly concerned with the FSO strategy and enterprise architecture (SEA) and service operation (SO) practice areas. These FSO areas use industry standard architecture practices such as TOGAF [50] for SEA and ITIL [51] for service operations (e.g. incidents, events and changes). The CiS architecture design process was overseen by the FSO SEA and OS areas. The CiS architecture was reviewed and certified by the SEA team to ensure that it is fit for the purpose, implementation and use in the FSO enterprise production environment.

4.3. ADR

The CiS was delivered over a period of 2 years by applying the ADR method. The ADR method is organised into stages: principles, practices (based on [25]) and CiS project artefacts. The insights from the application of the ADR method to the CiS are summarised in Table 3.

4.3.1. Stage 1: initiative

The project research problem was initiated by the FSO as a strategic initiative (practice driven). This research initiative had the sponsorship and backing of the FSO CIO, which is extremely important for the success of any project or initiative. In the initiative stage, research problem for the project was discussed during the research idea workshops and meetings with the FSO. This stage draws on the ADR

practice-inspired research principle (Principle 1) and practices (Table 3). The idea of developing the CiS was mutually explored by the FSO and UTX. The FSO had the known practical problem in hand but no known solution. One of the FSO internal review reports highlighted that '0 commercial off-the-shelf CMDB solutions available that meet project needs.' The decision was to develop the CiS architecture and use the MDA to build the majority of the CiS architecture components. However, some components could be purchased and integrated (e.g. data scheduling and enterprise social networking). Hence, UTX engaged to work with the FSO to co-create novel architecture design for implementing the CiS solution. The FSO is a sensitive public organisation of more than 1000 people; thus, a non-disclosure agreement (NDA) was also signed, and a secure communication channel (private email box) was setup to share information between the FSO and UTX. The key artefacts were project idea document and the NDA. The existing ADR methods do not explicitly specify this stage. However, this stage has been identified on the basis of the exiting ADR method principles 1 (practice-inspired research) [21]. This principle emphasises that the field or practical problems should be the focus as knowledge-creation opportunities (as opposed to theoretical puzzles). This research initiative stage 1 triggered the ADR project initiate stage 2.

4.3.2. Stage 2: initiate

This stage draws on the following two principles: practice-inspired research and theory-ingrained artefact (Table 3). The challenge for the FSO was the management of configuration information of a large portfolio of interdependent technology services. This practice-inspired research proposed the development of CiS for the FSO. The FSO strategy

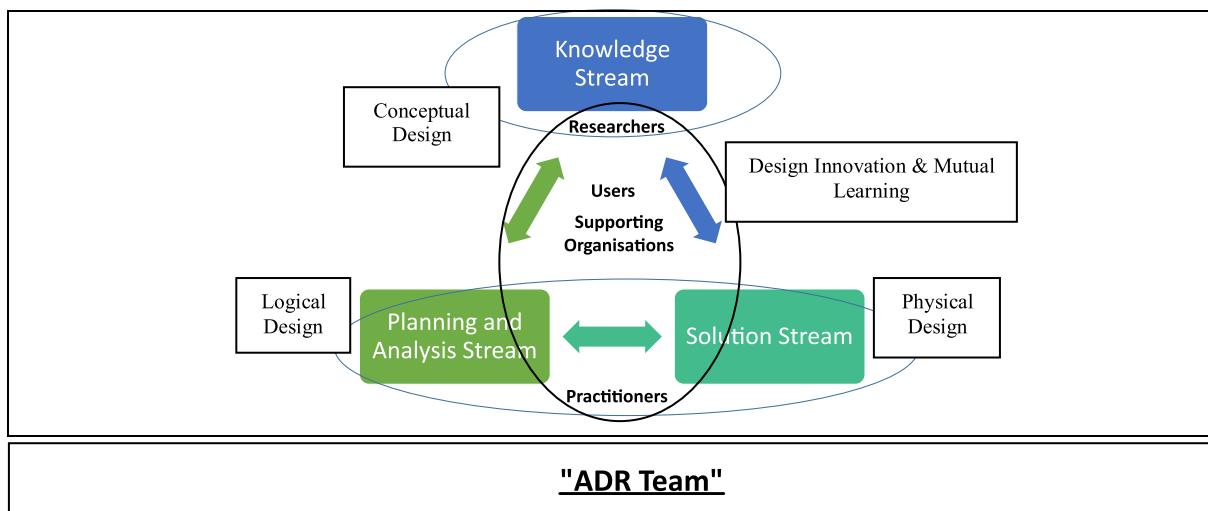


Fig. 2. ADR Project Work Streams (Value Co-Creation).

was reviewed to define the project vision and scope. The focus was on clearly detailing the project idea in alignment with the FSO strategy (critical for our research approach) and, correspondingly, clearly defining the ADR project problem and objectives (vision and scope). In this stage, a stakeholder analysis was performed, and the stakeholders who will support the project (e.g. sponsors), collaborate to get the project done (e.g. core team, working party and users), respond (e.g. external partners responding to any queries) and inform (e.g. need to inform about the project deliverables such as security team) were identified. Fig. A1 in Appendix A includes stakeholder analysis workshop photos as examples. Please note that the workshop photos are purposely kept blur because of the FSO privacy concerns. From the stakeholder analysis, three work streams were formed (e.g. architecture knowledge stream, planning and analysis stream and solution stream), which will be discussed later in this paper (see Fig. 2).

The project's vision was defined, and scoping workshops were conducted involving two UTX researchers and the FSO service management, architecture and implementation professionals and key users for the identification of high-level requirements. Our research strategy was to first crystallise the high-level requirements to understand the vision and scope of the project. There were more than 200 technology services in the FSO; thus, as opposed to a big bang traditional approach to the designing and implementation of the CiS, an initial list of six high-value services (e.g. email service, payment service and network service) and related repositories were identified at this stage.

The initial CiS solution was scoped to provide the ability to view service dependencies and asset details only stored in different systems, perform impact assessment and root cause analysis (e.g. incident and problem management) and identify trends or changes (e.g. change management) for services. The CiS solution (RestAPI driven) was designed to adapt and accommodate other services and integrate with the existing configuration and assess management repositories (sources of truth). The CiS solution needs to incorporate related FSO regulatory compliance requirements and enterprise information architecture principles (e.g. information has a common definition and information is secure). This stage determined the initial project vision and scope for defining the problem as an instance of a class of problem, which provided the impetus for formulating the plan, contract, funding, roles (work streams) and securing a long-term commitment from the participating organisation FSO for the next stage of the ADR.

A number of kernel theories (e.g. AESS and MDA) identified and reviewed with the FSO team to create theory-ingrained artefacts (Principle 2) for this project (Table 3). Vision and scope stage presented

the opportunity for scholarly knowledge identification and creation from this practice-oriented ADR. Here, one of the key learning is that the kernel theory of MDA was used as a guiding lens to define the CiS architecture-driven work streams structure into three areas (Fig. 2): knowledge stream (conceptual CiS architecture), planning and analysis stream (logical CiS architecture) and solution stream (CiS physical architecture)]. These streams were supported by the external FSO CiS users and partner organisations. The key artefacts from this stage were project vision and scope, architecture principles, project plan, project contract, work stream structure and kernel theories (Table 3).

4.3.3. Stage 3: discovery and DevOps

The integrated Discover and DevOps (development and operations) stage (Table 3) refers to the iterative building, intervention and evaluation (3a. BIE) and reflection and learning (3b. RL) of the CiS artefacts over a period of 2 years in small releases. Each quarterly release (3 months) includes fortnightly iterations similar to that in the FSO project delivery cycle and requirements. This stage draws on ADR principles of theory-ingrained artefacts (Principle 2), reciprocal shaping (Principle 3), mutually influential roles (Principle 4), authentic and concurrent evaluation (Principle 5), guided emergence (Principle 6) and relevant practices (Table 3).

4.3.4. Stage 3a: BIE

The BIE stage includes principles 2–5. This stage interweaves the iterative building of the theory-ingrained (Principle 2) CiS artefacts, intervention in the FSO organisation and evaluation (BIE) by following an iterative process appropriate to the organisational context (Principle 3). During BIE, the CiS architecture artefacts are continuously co-created (mutually influence) through active end user or stakeholder engagements and evaluated, and the design principles were emerged to address the research problem (Principles 4–5).

The BIE in the ADR method emerged as an incremental social design innovation process for the FSO. It was mainly an organisation-dominant BIE as opposed to an IT-dominant BIE. An IT-dominant BIE mainly focuses on creating the innovative technological design. Organisation-dominant BIE, as in this project, emphasises on innovations such as the CiS design for the FSO (organisational) research problem (class of problem) and also knowledge co-creation from this organisational intervention.

The ADR team or work streams at the FSO engaged in recursive adaptive cycles of design innovation and mutual learnings among the project work streams at conceptual, logical and physical architecture

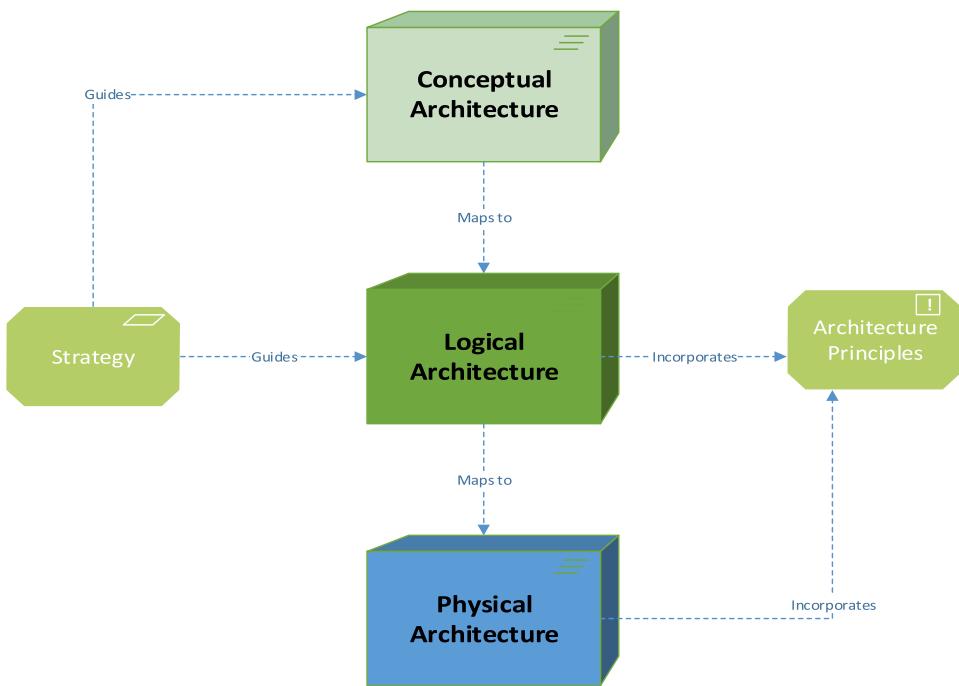


Fig. 3. CiS Architecture Layers (based on MDA Theory).

Table 4
Organisation-dominant BIE – FSO ADR Project Streams: Integrated engagement and governance.

Researchers (Knowledge)	Practitioners (Implementation)	Supporting Organisations (Implementation)	Users (Evaluation)
CiS Conceptual Architecture Alpha	CiS Logical and Physical Architecture Alpha	Solution Development Support	End User Testing
Beta	Beta	User Experience Design (UXD)	
Gama	Gama	Support Architecture Review	

Shared Area: Review, Feedback & Adjustments.

design layers (MDA layers based on architecture kernel theory) (see Figs. 2 and 3). The researchers from UTX largely contributed to the conceptual design and mutual learnings through their knowledge of design theory and technological advances (knowledge stream – Fig. 2), while the FSO practitioners including supporting organisations largely contributed through their practical knowledge of FSO work practices and the environment in which the FSO operates (integration of research and practice). Users were involved in all aspects of this process. Further, BIE is not a linear stage-gate model. In BIE, artefact building and intervening are interwoven, and evaluation is an ongoing (concurrent development and evaluation) activity and is not an isolated activity, which follows the building activity [43]. The knowledge stream (researchers) was mainly responsible for the conceptual CiS architecture model. The planning and analysis stream (practitioners) was responsible for turning the conceptual architecture into the logical CiS model, and the solution implementation stream (practitioners) was responsible for turning the logical CiS model into the physical CiS model or implementation (coding and infrastructure) (Table 4). These three streams also include users and supporting organisations. These streams worked together to iteratively co-create CiS models (artefacts)

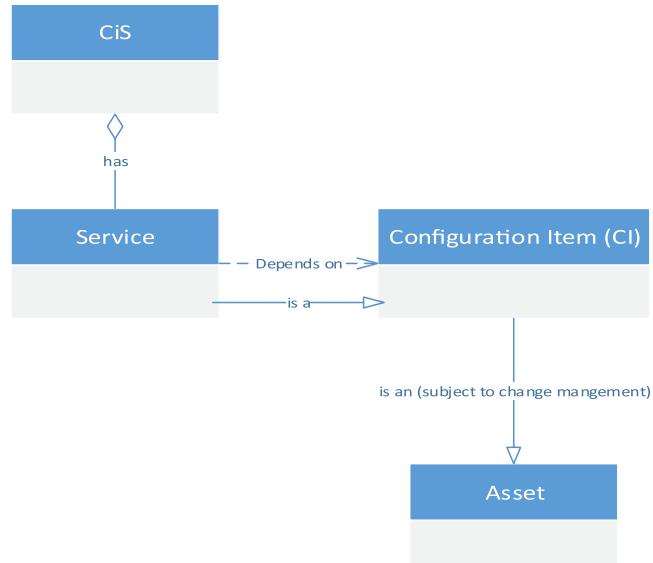


Fig. 4. CiS Architecture Context Model (Level 0).

via review, feedback and adjustment cycles. These architecture layers were continuously reviewed to ensure alignment to the FSO strategy and existing FSO enterprise architecture principles (FSO architecture governance).

Fig. 4 provides the CiS architecture context model (level 0) for the FSO, which highlights the three major architecture building blocks of the CiS (Fig. 4): service, CI and asset. This model shows that a service depends on CIs and is also a CI. A CI is an asset, which is subject to configuration management (e.g. change impact analysis and root cause analysis). These three building blocks were detailed in terms of conceptual (level 1), logical (level 2) and physical models (level 3). Here, we present the details of service-related models as examples.

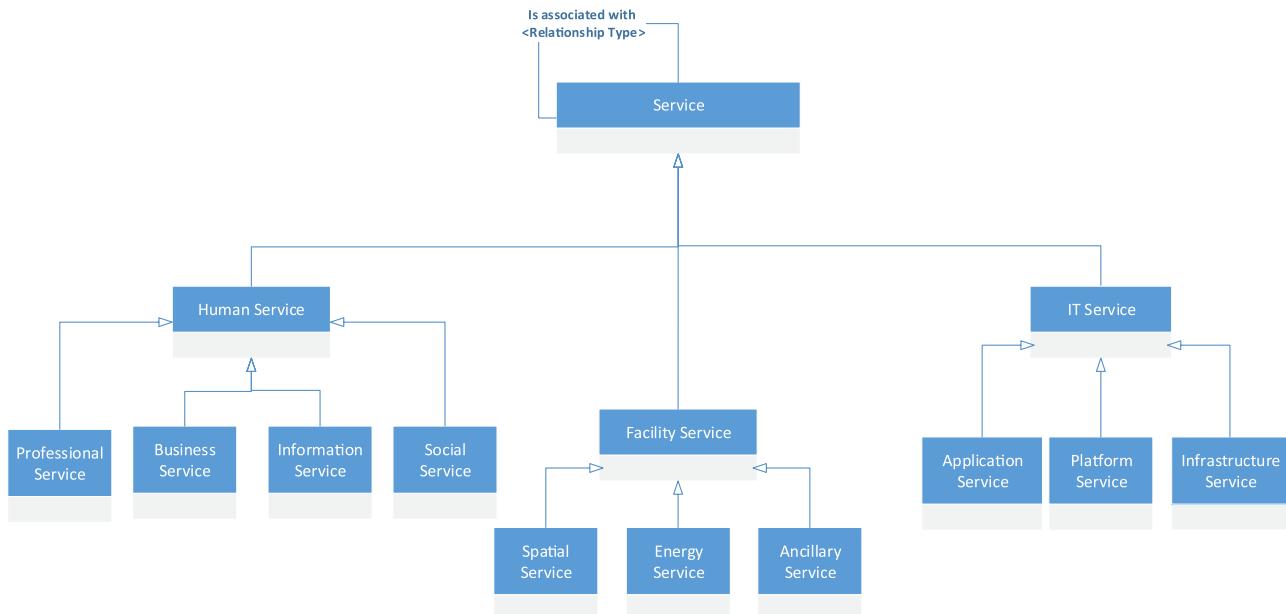


Fig. 5. CiS Conceptual Architecture Model (Level 1: Service Classification) (based on AESS metamodel [46]).

4.3.5. Conceptual architecture

The CiS architecture is based on the theoretical AESS metamodel [25,26] (theory-ingrained artefact). In practice, an architecture is

designed using some relevant reference or metamodel; thus, in this project, we used the service-oriented AESS metamodel as a reference model to develop the CiS architecture for the FSO context. This

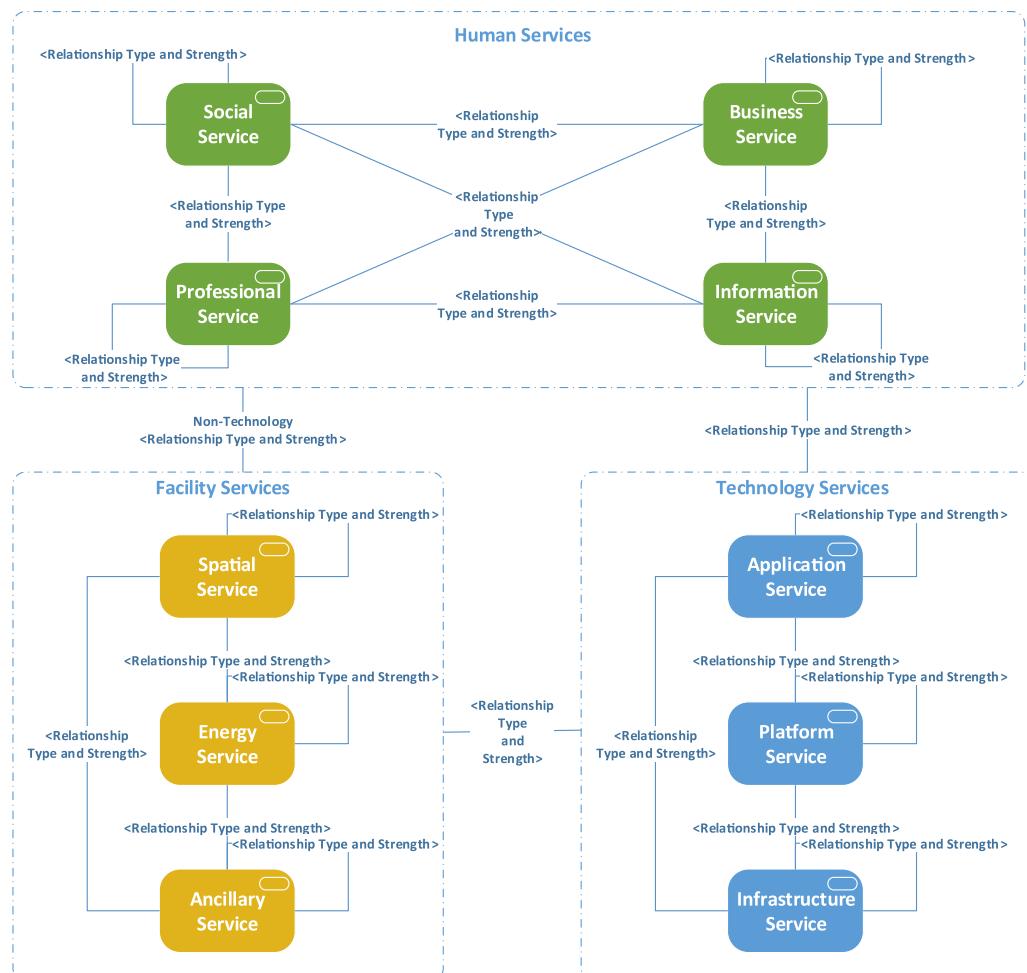


Fig. 6. CiS Conceptual Architecture Model (Level 1: Service Relationship) (based on the AESS metamodel [46]).

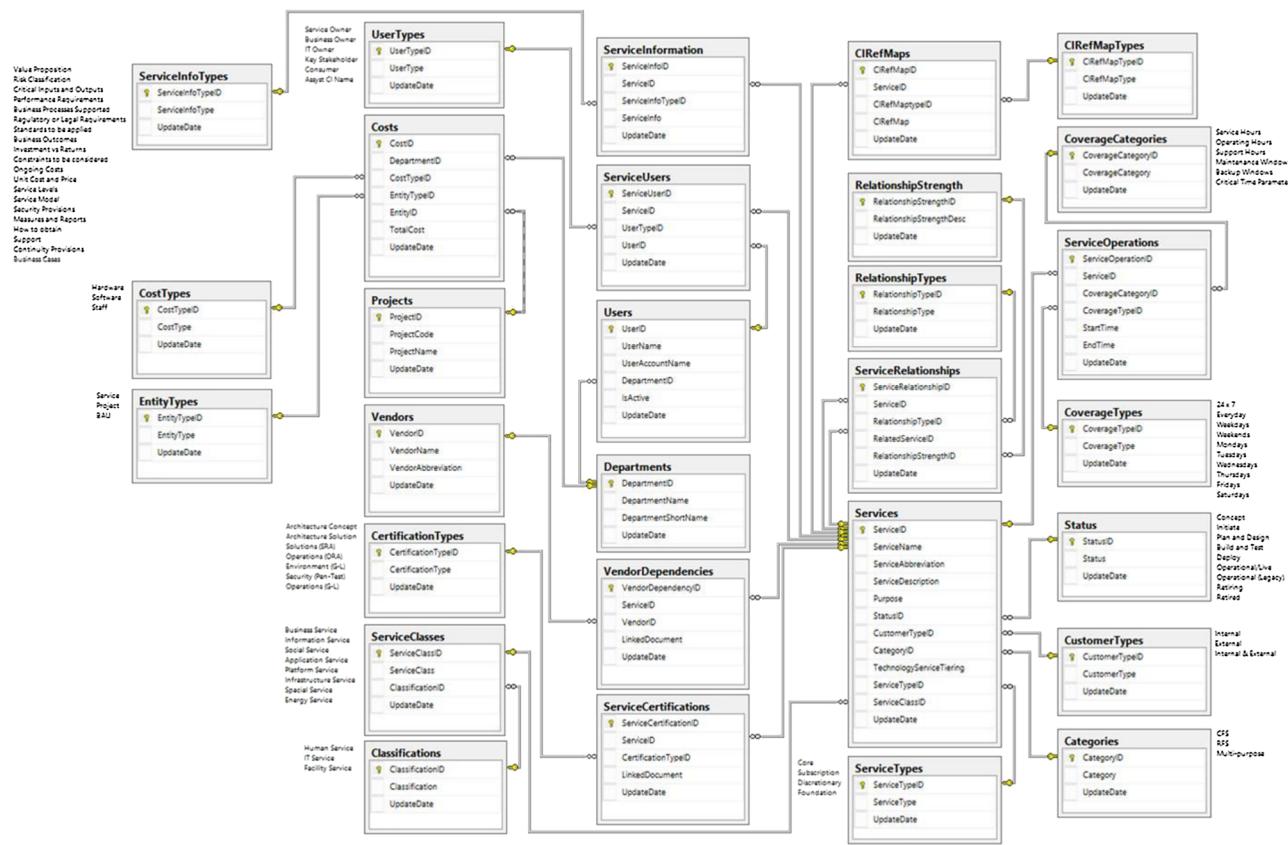


Fig. 7. CiS Logical Architecture Model (Level 2).

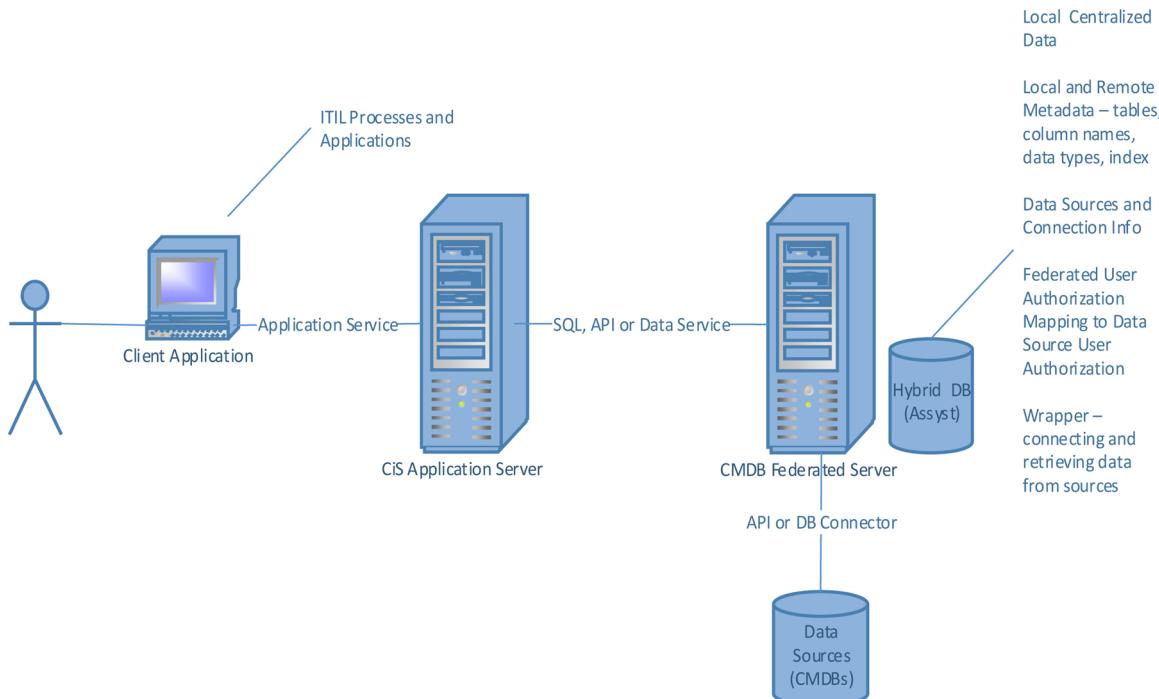


Fig. 8. CiS Physical Architecture Model (Level 3).

metamodel was used because it specifically offers generic service-focused architecture elements, which provide the foundation to build the service-focused CiS architecture. This section discusses the CiS conceptual architecture models for FSO services.

A service is the application of resources (e.g. Cls) for delivering

value to internal and external customers by facilitating outcomes the customers want to achieve without the ownership of specific costs and risks. There are two types of conceptual service models at level 1: service classification model (Fig. 5) and service relationship model (Fig. 6).

The service building block is core to the CiS architecture and was classified in terms of human, IT and facility services. Human service is further classified into professional, business, information and social services. Similarly, IT service is classified into application, platform and infrastructure services. Facility service is classified into spatial, energy and ancillary services. These services were further classified (e.g. customer-facing service, resource-facing service and multi-purpose service) as the conceptual architecture was iteratively evolved from alpha to gamma version over a period of 2 years.

The service relationship conceptual model described the relationship between these services (Fig. 6). A service can be associated to itself (self-relationship), with another service in the same layer or level (peer-to-peer) and with another service in a different layer or level (cross-layer hierarchical).

4.3.6. Logical architecture

The logical architecture details the conceptual architecture building blocks. Fig. 7 details the service classes, attributes and their relationships independent of technology implementation. For instance, it shows the service class, its attributes and relationships to other classes such as service types and categories and their relationships. One of the key classes is service relationships, which capture the information about relationships between the FSO services. This class diagram was then converted into a database during the physical architecture design stage for storing the information about services and related entities as outlined in Fig. 7. Similar to conceptual architecture, logical architecture evolved from alpha to gamma (final version) through BIE cycles.

4.3.7. Physical architecture (implementation)

The physical architecture is the implementation of logical architecture (software and hardware solution and products for storing and retrieving service configuration information). A number of options were evaluated through BIE workshops, and proof of concept was developed to arrive at an appropriate physical architecture. A cached federated approach was devised to design the physical architecture. There were over 25 existing data sources or repositories for storing information about services for the CiS. In a cached federated physical architecture, the CiS metadata are centrally stored, and actual data are not moved into the large CiS monolithic database. The actual data are stored in the source system and retrieved from the integrated sources and loaded into the fast cache memory of the CiS system for frequent and fast access in case data source system is down or not available for any reason. This approach resulted in the best performance of both federated and cached architectures.

The CiS architecture implementation was done using the model, view and controller architecture (MVC) pattern (see Fig. 8). It is composed of a typical web client application (view), application server (control logic) and federated CMDB server (physical model or data layer: CiS database) and underlying source systems or repositories. The client application accesses the service-related data from the federated server through controller logic or application server. This ensures that client application is independent of data layer. This was done to ensure that in future, the architecture can easily adapt to support other applications (multichannel access) such as access of the data by mobile client applications without changing the data layer. The federated data layer provides the seamless integration and access to service-related data stored in different repositories. This architecture was evaluated in conformance with the FSO architecture principles and industry best practice, and it successfully obtained design certification and approval through external review by the FSO vendor and FSO strategy and enterprise architecture governance team. This ‘principles and industry best practice’-based evaluation ensured that the design meets the design principles and requirements, is fit for purpose and secure, is developed according to the approved FSO process, applies relevant industry best

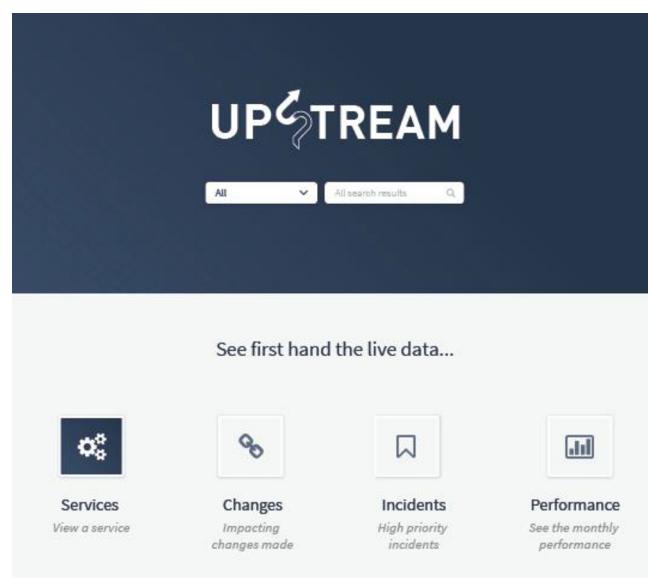


Fig. 9. CiS Landing Page View.

practices (design patterns such as MVC), and is appropriately documented. For instance, the CiS architecture was evaluated against the principles of ‘ease of use,’ ‘simple light-weight data model,’ and ‘cached federated model’ etc. (please see Table 4 for design principles). The initial design certification was obtained at the early stages of the project for the conceptual design. Later, the design certification was obtained as the design emerged from conceptual to physical architecture. It is important to note here that for the FSO to put any system such as the CiS in production environment, it is mandatory to go through the rigorous architecture design certification process and external design peer review. The CiS client application was implemented using the Angular JS (java script), which is quite mature and extensively used for web-based applications in industry. ASP.MVC 5 (industry best practices) was used to implement the application server as it meets the needs of the CiS data-intensive solution in terms of required integrated data access in a secure manner. Finally, the CiS federated database was implemented using the relational SQL server database engine.

Figs. 9 and 10 provide the landing page and service map or dependency view examples. These figures show the implementation example of conceptual and logical architecture models depicted in Figs. 6 and 7. For privacy reasons, these figures have been blurred. The service landing page shows that the CiS can upstream the data from different repositories (e.g. service incident, change and performance data). The solution is live and evolving since we completed the project. The live view of services and its related data are displayed on a big screen outside the CIO office.

4.3.8. Architecture evolution (alpha to gamma)

The CiS architecture layers evolved in terms of alpha, beta and final gamma versions.

- Alpha (Initial version)
- Beta (Intermediate versions)
- Gamma (Final version in production)

Instead of a big upfront detailed conceptual or logical or physical architecture, as would traditionally be seen when waterfall methodologies are used, it was decided to begin with just enough conceptual architecture (alpha version). This was done to enable the conceptual architecture to evolve (from alpha version to the final gamma version)



Fig. 10. CiS Service Map View.

as the CiS conceptual architecture is implemented (logical and physical models) in small releases through the user feedback loop mechanism (see Fig. 2). Alpha versions comprising of all three layers were designed to develop the initial proof of concept in release 0. This helped us to understand the complexity of the CiS architecture at all three layers at the early stages of the project. A number of architectural decisions were made on the basis of the learnings from the development of proof of concept. For instance, for the conceptual architecture, it was decided to keep the conceptual classes of services and underlying assets at the minimal number level and avoid over engineering of the conceptual model but make it adaptable to allow the model to add new conceptual elements (e.g. services and repositories) in subsequent releases for different user contexts (e.g. incidents, events, changes, problems, performance and capacity). For the logical layer, it was discovered to only start with attributes that are most important from user perspective

(early user engagement). Do not create class for items that are really an attribute. For the physical layer, it was decided to implement in-house existing development capability and technology stack. Data will be stored in the source systems, and any spreadsheet will be converted into a relation database as they are discovered during project iterations. Data movement from the source to federated server will be minimal. As discussed earlier, the MVC architecture was used to implement the CiS alpha, beta and gamma versions of the CiS solution. Data were exposed via JSON using RESTAPIs and https and visualised using AngularJS (Java Script).

Building on the initial proof of concept, the remaining releases developed the beta intermediate versions of the CiS. Gamma is the final version 1.0, which was released in production in the end of the last release for formal use by the users. The conceptual CiS architecture was implemented (logical and physical architectures) using the practices

Iterations/ Sprints

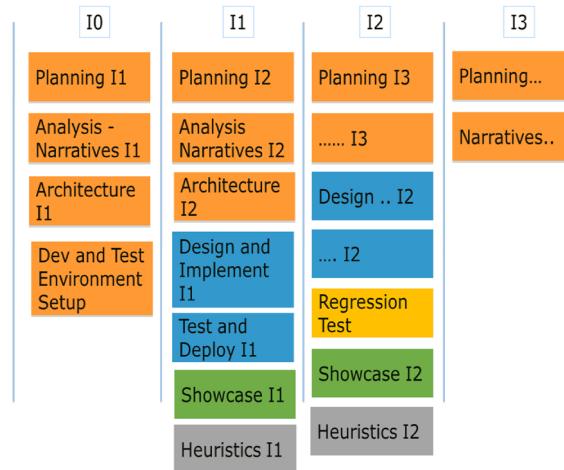


Fig. 11. CiS Iteration Orchestration.

from agile and non-agile implementation and service management body of knowledge. The adaptive enterprise project management (AEPM) capability reference model was used to inform the iterative implementation of the CiS in small releases. The AEPM reference model layers and services are shown in Fig. B1 in Appendix B. The implementation was also supported by external organisations as indicated earlier (Fig. 2 and Table 4). Each quarterly release (approximately 3 months) has a number of iterations (2 weeks) and is planned to deliver a set of high-priority services and related user stories and NFRs, which are related to the CiS architecture. Each fortnightly iteration (a.k.a. sprint) involves detailed iteration planning, analysis, architecture, design, implementation, testing and deployment of selected user stories. Each iteration is aimed at producing the CiS solution increment, which is showcased to product owner and related stakeholders. Each iteration involves heuristics (retrospective and adapt) session to reflect and adjust the CiS solution process. Each iteration or sprint is divided into three parts.

- Pre-iteration/ Sprint
- Iteration/Sprint
- Post-iteration/ Sprint

Pre-iteration involves planning, analysis and architecture details confirmation for the upcoming iteration. For instance, planning, analysis and architecture for iteration 1 are confirmed in iteration 0. Planning, analysis and architecture for iteration 2 are confirmed in iteration 1. The idea is that the analysis and planning team should be at least 1 iteration ahead of solution implementation team to avoid any blockers. Actual design, implementation, testing and deployment are done in main iteration or sprint. The developed solution is showcased to users to get feedback in each iteration. The solution team reflects and adjusts in post-iteration heuristics (retrospective and adaptation). Fig. 11 presents the details for the CiS iteration orchestration process. I0 column (Fig. 11) presents iteration 0, which involves planning, analysis (user story or requirement narratives) and architecture for iteration 1 of a release. I0 also involves setting up of development and test environment before the development of iteration 1 (I1). I1 involves the design, implementation, testing and deployment of iteration 1 user stories or requirements. Iteration 1 solution is showcased to users for feedback. Heuristics or retrospective is used to improve the development process. It also involves the planning, analysis and architecture of the next iteration, I2. Similarly, other iterations can be seen in Fig. 11, which

indicates the evolving nature of the CiS solution in small iterations. A typical release has 6 iterations in total.

In summary, the BIE stage was mainly focused on iterative building, intervening and evaluating the CiS conceptual, logical and physical architecture design artefacts. The next stage discusses the reflection and learning (RL).

4.3.9. Stage 3b: RL

Active and continuous feedback helped to continuously refine the CiS architecture. New features emerged as we progressed through this feedback-based continuous reflection and learning (RL) mechanism [Principles 6]. ADR is not only about solving a particular problem but also about reflection and learning, which was the first step in the generalisation of the CiS artefacts (e.g. Fig. 3). In RL, practices such as stakeholders' feedback, review workshops or showcases, retrospective and resultant continuous design refactoring reflect the increasing understanding and evolution of the ensemble CiS artefacts and process (as indicated in principle 6: guided emergence).

4.3.10. Stage 4: close

Finally, the close stage was focused on formalising the learning and final feedback at the end of the project. This stage draws on the following principle and practices of ADR.

• Close (August 2016)

- Principle
- Principle 7: Generalised Outcomes
- Practices
 - Abstract learning into concepts for a class of field problems
 - Articulate outcomes as design principle
 - Formalise results for dissemination
 - Feedback and final payment

The objective of the fourth stage was to formalise the learning for generalised outcomes (principles). Thus, this ADR project further discovered general design concepts and principles (see Table 3 and Figs. 2–6). For instance, work stream structure and conceptual models can be applied to other ADR projects with the aim of addressing a different class of problems. Casting the ADR learnings into a class of problems is important to facilitate the practice-oriented research knowledge contribution and conceptual change.

The following section further expands on the learning for generalised outcomes in terms of ADR implications and proposes the design principles, which is a key contribution of this ADR project both from research and practice perspectives.

5. Discussion: challenges and implications

The CiS was designed and implemented to manage the configuration information of a large number of FSO services. The CiS design and implementation were not straightforward tasks. They have both challenges and implications (research and practice).

5.1. Key challenges

We faced a number of challenges during the CiS architecture design and implementation project, which are discussed here.

First, we needed to identify the fundamental unit of design for the CiS architecture. There were at least two main options: service or service resources (e.g. service components: hardware, software and information). It was decided that a service is a fundamental unit of design in the CiS architecture. This is because the customers or users are more concerned about the service (e.g. service performance, service quality and service availability) as a whole rather their individual parts or

components. Thus, a service-oriented approach was selected to design and implement the CiS architecture.

Second, there were over 200 technology services, thousands of underlying components and over 25 service repositories. The challenge was where to begin? We identified, prioritised and began with high-value or important services and relevant repositories from customer or user perspective. A series of service design discovery workshops were organised at the FSO to iteratively design and develop the service element artefacts for the CiS.

Third, services and related data were stored in the source systems (service data repositories) and a number of spreadsheets. The challenge was how to integrate spreadsheet data for the CiS? It was decided that all the spreadsheets will be turned into relation databases as they are discovered during project iterations.

Fourth, the challenge was how to classify and catalogue the large number of services at a higher manageable level for the CiS? Thus, a service conceptual model (Fig. 6) was developed (based on [46]) and used to catalogue and classify the FSO services in terms of human, technology and facility types of services and their relationships (type and relationship strength). Fig. 6 shows that a service can self-reference and has relationship with other services. Fig. 7 details the logical model that comprises of service classes and attributes to manage the information of FSO services across humans (business, social, information and professional services), technologies (application, platform and infrastructure services) and facilities (spatial, energy and ancillary services). Figs. 8–10 present the physical implementation models. These models represent the service-oriented CiS architecture.

Fifth, the challenge was that which services should be included in the CiS. It was decided to include only those services that are in operations and are being used by the users. Each service once is deployed in operations will be included in the CiS. This is to avoid the cost and effort in managing unnecessary service information. Thus, it is important to mention here that this research only addresses the services in operation. Others may study the services in pipeline (not yet deployed in the operational production environment), which is beyond the scope of this paper and research project.

Sixth, the challenge was the alignment of researchers and practitioners' ways of working. Thus, in the beginning of the project, we defined a common way of working by aligning the researchers' ADR and the FSO's own architecture and development approach. This helped us to avoid or minimise the conflict between these two main parties of the project. During the project, the main focus was the delivery of artefacts as opposed to a paper publication. This is the reason why we compiled this main paper after the completion of our research project.

5.2. Implications

Literature on the CiS in particular financial services is scarce. This paper contributed to research and practice by filling this small research gap and reported research and practice-based insights from an ADR project of the CiS in the FSO context. The main research question was as follows: how to design and implement the CiS architecture in the practical context of the FSO? This paper reported the ADR method and key CiS artefacts as generalised outcomes as a contribution. This section discusses a number of important implications.

5.2.1. Research implications

One of the key contributions of the design research is to offer research implications in the form of design principles. This section highlights the design principles extracted from this practice-oriented ADR project. These principles can be adopted to design and implement CiS-type architectures and class of problems for similar context.

First, to effectively deliver such a large research project (spanning over 2 years), we needed a team of motivated individuals. It is not

about the processes so much, but it is about the people or team who actually collaborate and co-create value (design and knowledge about design). Therefore, in the early stages of the project, a collaborative and motivated cross-institutional team comprising of people both from academia and professional organisations was formed to co-create the CiS artefacts. This indicates the creation of an integrated engagement and governance model (work streams) for value co-creation and performance tracking. This also provides a link to the concept of design communities from digital infrastructure [53].

Second, it is important to understand the different levels of stakeholders and their needs such as strategic (e.g. Operations Senior Manager), tactical (e.g. Operations Managers) and operational (e.g. FSO Service Operations Staff). Thus, a great attention was given in the early stages of the project to carefully identify and analyse the project stakeholders and their needs. Thus, we need to involve stakeholders early and continuously in the ADR project. Social innovations often affect the social practices of stakeholders; therefore, the involvement of end users in an early stage is recommended, even before any alpha or beta versions of the CiS are produced. This reflects our stakeholder or customer-centric approach of ADR.

Third, by reviewing the traditional CiS-type systems, it was found that these systems are complex in nature and not easy to use. Therefore, the FSO emphasised the ease of use principle for the success of the to-be developed CiS architecture and system. A CiS is a complex system that involves more than 25 data repositories at the FSO that interact with each other and other systems. It was clear to have a simple and flexible light-weight common data model to reduce the complexity and development effort. This is linked to the dynamic and inter-connected characteristics as explained in the digital infrastructure literature [53].

Fourth, the FSO senior managers emphasised the design principle of adaptability. Thus, the CiS needs to be able to adapt to further or new data repositories; therefore, an adaptive data model (e.g. accommodate new data sources, service classes and attributes) was designed for the CiS architecture. Thus, the core of the CiS design was around the federated model as opposed to the traditional all in one big inflexible large data repository. Data will remain at the source and can be loaded into the cache through metadata links, which will enhance the availability of the data from the cache even when the source system or repository is down. The federated cache data model enabled us to avoid the unnecessary movement of data between the CiS and the source systems or repositories. This is to minimise any possible errors or link down errors between the source systems and the CiS. The adaptability and flexibility can be traced back to digital infrastructure characteristics [53].

Fifth, the CiS was designed around read-only access to the service data stored in different repositories. The read-only basic service information view provides the ability to drill down when required to view detailed service information from the source repository. Operations manager proposed that the CiS should provide a window and single view to complex integrated data such that the underlying repositories are transparent to the end user.

Sixth, reciprocal shaping between social communication and collaboration practice and artefacts is another principle to be considered. Design cycles were iterated between shaping the CiS artefacts and the affected social communication and collaboration practices at the FSO. CiS artefacts led to ideas on how to improve practices of operational resilience of the stakeholders involved, and vice versa. This also provides a link back to the digital infrastructure concept that highlights the social embedded IT artefacts, sociotechnical world and coordination [53].

The FSO is a highly sensitive organisation; finally, we attempted to understand and balance political, economic, privacy and social values of the FSO when evaluating and interpreting the ADR results. As an ADR community of practice, we should be well aware of the different political, economic, privacy and social values that play a role in social

Table 5

Key design principles emerged from the ADR project.

Traditional CMDB/CiS Challenges	Design Principle	Description: Guiding Statements and Recommendations
Vendor driven	Build team around motivated individuals to address the CiS as a business problem (business-driven value co-creation)	Build a motivated cross-institutional team comprising of people both from academia and professional organisations to co-create the design artefacts with a view to address business problems. It is not solely the responsibility of the engaged ADR researchers to design the technical artefacts while ignoring the business problem.
Solution driven as opposed to stakeholder driven	Translate a practical problem on different stakeholder levels	Interpret the practical problem from different stakeholder perspectives (as opposed to solution imposition) at different levels and sought executive sponsorship and ownership for the project (the CiS project was initiated by the CIO).
Difficult to search/find things Over designed/engineered schema	Easy to use Simple light-weight data model	Customer-centric, easy-to-use architecture design and resulted system. A simple light-weight data model is required for the CiS to aggregate and cache data from different data repositories; this can result in faster seamless data retrieval and less reliance on data sources for event-based APIs and CiS availability.
Lack of adaptability and integration Heavy centralised storage	Adaptive data model Cached federated model	Built-in adaptability in the architecture design so that it can accommodate and adapt to future services and user requirements. Cached federated model for CiS-type systems will enable the users to get the data from the local cache at a high speed even when the source data repository is not available or down.
Data governance	Read-only basic service information view with the ability to drill down	Source systems that feed data to the CiS will implement and ensure data governance, and the CiS should only provide a read-only view and links to the data stored in the federated repositories. Scheduling was required to periodically poll data from different sources and store in the CiS data cache without relying on the availability of data sources.
Data integration	Underlying data stores are transparent to users	CiS-type systems need to seamlessly integrate with the underlying source repositories or systems for smoother user experience.
Data quality (Garbage in, garbage out and/or stale data)	Avoid unnecessary data movement	Data movement requires more integration and management of data movement links and duplications. Avoid unnecessary data movement to ensure the consistency of data.
Data quality (Garbage in, garbage out and/or stale data)	Built-in-feedback capability	Built-in feedback capability to allow users to provide updates and feedback on the accuracy of data.
Lack of continuous improvement	Reciprocal shaping between social communication and collaboration practice and artefacts	Design cycles iterated between shaping the CiS artefacts and the affected social communication and collaboration practices. CiS artefacts led to ideas on how to improve practices of operational resilience of the stakeholders involved, and vice versa.
Lack of stakeholder active engagement	Involve stakeholders early and continuously in the ADR project	Social innovations often affect the social practices of stakeholders; therefore, the involvement of end users in an early stage is recommended, even before any alpha or beta versions are produced.
Lack of a balanced approach	Balance political, economic and social values for evaluating ADR results	The ADR researcher should be well aware of the different political, economic and social values related to the organisation that play a role in the social innovation.

innovation. Table 5 presents the summary of design principles emerged as a generalised outcome (principles 7) from this ADR project.

5.2.2. Practical implications

ADR as an innovation process for industry to tackle complex problems such as the CiS through collaboration between academia and industry. An effective architecture design and implementation of the CiS-type systems is challenging. The traditional architecture and implementation of the CiS as a large centralised CMDB that involves the big bang technology-oriented approach, where focus is more on identifying, sourcing, processing and managing technology asset information such as CI management. In a normal medium to large enterprise, there are hundreds of technology services and thousands of technology-related CIs such as software and hardware (applications, platforms, computers, routers, sensors, mobile, servers and firewalls). The information about these CIs is stored in different dispersed repositories and spreadsheets. A big bang approach to identify source, process and manage such a large volume of information is very time consuming and neither appropriate nor practical. It has been noted that Industry has tried and failed in the effective implementation of the CiS using the big bang technology-oriented approach.

The FSO engaged UTX researchers to research and define an alternative approach ADR to effectively design and implement the CiS

architecture. This demonstrates the value of using the ADR for designing and implementing the service-oriented CiS architecture. A number of observations and implications for practice have been made.

First, the CiS is not an operational one-off project, and the development of such a large complex system is not an easy task. It is a strategic initiative, and we need to define an agile or adaptive thinking and approach for iteratively designing and implementing the CiS architecture. This demonstrated the incremental research benefits to the participating FSO organisation where short iterations (fortnightly: 2 weeks of duration) and releases (quarterly: 3 months) trigger the benefit-response reaction, thus leading to more engagement and higher quality result.

Second, there is a need for service-centric (service science) living systems and model-driven architecture to design and implement the CiS. The focus, at the initial high level, should be on the identification of the enterprise business context and high-value services (such as the business services and technology services) for the CiS.

Third, the CiS is a large initiative. It should be managed and broken into small releases. Each release should focus on the delivery of a small set of selected high-priority services for the CiS architecture (see Fig. B1 in Appendix B). This can be shown to have alignment with human productivity principles where small short-term goals trigger the benefit-response reaction, thereby leading to more engagement and higher

quality results [52]. For instance, select one of the identified high-priority services and identify, source, process and manage the important information about that service including its CIs and sources of truth (data repositories) in defining the CiS. This iterative approach is more business centric and less risky in contrast to the traditional technology-focused big bang approach. Thus, instead of ‘boiling the ocean,’ we should focus on a piecemeal service approach.

Fourth, in a high-level CiS architecture development, as opposed to a big upfront detailed architecture, the CiS architecture will evolve as the CiS solution is emerged through implementation in small project releases and iterations. In a traditional environment, enterprise architecture capability is operated in isolation of project implementation capability. The ADR process integrates these two isolated capabilities and enables the active engagement of the enterprise architecture capability (the CiS architecture design) and implementation capability (architecture implementation).

Fifth, an organisation that encompasses the CiS is a complex undertaking involving a significant number of stakeholders having complex and conflicting requirements. Therefore, the focus should be on co-creating the CiS through interactions between researchers and practitioners. In a nutshell, the stakeholders actively engage and participate in the workshops for co-creating the CiS. According to the value co-creation concept (adopted from the service science body of knowledge), the focus shifts from CiS delivery to CiS co-creation. The value co-creation concept guided to leverage the knowledge (e.g. architecture, design thinking, service management, agile project implementation practices etc.) of practitioners (FSO) and researchers (UTX) for co-creating both practice and theory-based robust CiS.

Sixth, it is clear from this research project that the ADR is an outcome-driven practice research approach and is a way to link practice (professionals) and research (academics) effectively. This project also highlighted that kernel theories, existing FSO artefacts and principles can be useful in combination with the ADR. The FSO mentioned that *‘has enabled us to approach the problem space in a different way more akin to the agile processes used by high performance software development houses than traditional corporate design-document-code-deliver phased processes.’*

Seventh, the FSO appreciated the theoretical aspect of the systems thinking and mentioned that ‘The theoretical work into Living Systems and ‘systems in systems’ is particularly relevant to the issue we are seeking to address in the design and development of our Configuration Information System as we are striving to deliver a system to support and encourage the resilience of the IT systems within the organisation.’ It is also important to note here that the FSO CIO executive’s ownership and sponsorship motivated the team and gave them some ‘intellectual’ power to probe deeper the FSO environment.

Finally, a post-implementation review of the CiS was conducted by using a feedback questionnaire with the FSO (through the CIO) in April 2018. The FSO mentioned that the CiS is *‘a part of ... of our operations management practice today.’* The CiS screened outside the CIO office provides the real-time information about services and their status. Further, the CiS is installed at other office locations to provide the users with the up-to-date information about services. The FSO mentioned that *‘screen outside CIO office [and at several other locations in the office building] to publicise the up-to-the-minute production release schedule to stakeholders who may be impacted by the releases.’* Further, the FSO reported that *‘It is the vehicle for our production release management and incidents management [as part of resilience management].’* The post-implementation review resulted in the following next steps from the FSO.

- *‘The first natural extension is to link it to asset management system [to streamline the management process].’*
- *‘The second extension is to add a predictive analytics system to enable real-time systems resilience management for the organisation.’*

This plan clearly reinforces the fundamental nature of the CiS to the organisation’s resilience management strategy.

5.2.3. ADR method implications

The ADR method, in alignment with the FSO way of working, was applied to the delivery of the FSO project. It is also perceived as a research contribution as it demonstrates the real life application and real value contribution of the ADR research principles and practices as suggested by many other action design researchers [12,25–28]. Insights from the ADR method perspective are outlined here.

First, we identified the practice of establishing a formal contract, which is often not discussed in the usual ADR method descriptions. This is very important because the contract contains important information for both parties, which need to be clarified to setup and manage the expectations upfront. This helped to maintain the collaborative relationship between the involved parties in a formal manner to avoid any possible disputes and conflicts of interest, which may lead to court cases and other unnecessary situations.

Second, in addition to a formal contract, we also signed a formal NDA. This is critical for the ADR project because of the active involvement of researchers with the professional organisation on a daily basis; this may expose confidential information and IP to both parties. The practice of creating such agreement has not been intensively mentioned in the traditional ADR method. The FSO is a very sensitive organisation, and careful approach is adopted to handle information disclosure and publications.

Third, the ADR project focuses on value co-creation and mutual benefit to all the involved parties, which are both the University and the Professional organisation. Thus, mutual benefit involved the creation of at least two types of artefacts: practice artefacts and research publication artefacts. Practice artefacts are targeted for professional organisation, whereas research publication artefacts are targeted for researchers. In this way, the ADR project does not only deliver what is required by the professional organisation (e.g. CiS architecture artefacts) but also what is expected from the research organisation (e.g. learning and publications). This is important to manage the two types of deliverables, which is sometimes difficult because of the conflicting performance priorities of practitioners and researchers. It is important to note here that the ADR project has not fallen into the trap of pure consulting or pure research where one of the parties may suffer from dissatisfaction and disputed relationship. There is a need to balance the resources across these two types of artefacts, which of course is challenging.

Further, it is important how the research funds will be paid to the University. In this case, a pay as you go model of payment was adopted where the payment was attached to the outcome or value in terms of research outcomes (CiS artefacts). The pay as you go model was tied up well with the adaptive incremental delivery of the CiS artefacts as opposed to a big bang delivery. This practice worked well for this ADR project. This indicates that the ADR needs to be adapted to different research and organisational contexts.

Occupation health and safety (OH&S) was carried out at the participating organisation site to ensure the health and safety of the researcher working away from the university at the partner site for a long period of time. This is another critical practice, which is often overlooked by the ADR project, and hence is proposed to be included in the ADR method.

Further, in addition to OH&S, researchers were insured by the University, though they were working at the participating organisation site. This practice is important to eliminate any risks arising from any kind of undesired situation including the impact on individuals and involved organisation. The insurance component here refers to the professional indemnity insurance, which is very similar to that in consulting projects. While the researcher was at the participating

organisation site, there was a famous Lindt Café attack or incident happened in Sydney on the same day. The researcher was stuck at the site building near the café. This reinforced how important is the issue of insurance and OH&S.

Finally, the publication of the ADR results required the approval from the participating organisation and University. This practice is again not well explained in the ADR project. This is important from the research perspective and needs to be explicitly mentioned, planned and executed in the ADR project. In this project, we needed to have ethics approval from the University as a whole and on each publication basis from the participating organisations communication unit. Thus, this has been proposed to include in the ADR method as an important practice.

6. Conclusion

This paper demonstrated the application and importance of the ADR method for designing and implementing the CiS architecture. This paper demonstrated how to use the ADR with kernel theories, existing artefacts, principles and industry best practices for creating CiS-type artefacts in small releases and iterations. The benefit of using this approach is that it provides a service-oriented, holistic and systematic top-down strategic approach to continuous realisation in the context of incremental CiS architecture design and implementation. The research

presented in this paper has research, practice and teaching implications. The ADR approach and design artefacts discussed in this paper can be used and extended by both practitioners and researchers as appropriate to their local context. In future, we intend to further report the learnings to community from our ongoing action-design research at the FSO.

Acknowledgments

We wish to extend our sincere thanks to FSO for providing financial support for this industry-academia collaboration action-design research project. In particular we would like to thank CIO and senior managers Mr. David Kricker and Mr. Geoff Bird from the FSO for providing the necessary support and active engagement during this project. Please also note that any views expressed in this paper are those of the author/s and not necessarily those of the FSO. Use of any results from this paper should clearly attribute the work to the author/s and not to the FSO. All the information contained in this paper is provided for educational and informational purposes only. No responsibility can be taken for any results or outcomes resulting from the use of this information. While every attempt has been made to provide information that is both accurate and effective, the authors do not assume any responsibility for the use or misuse of this information.

Appendix A

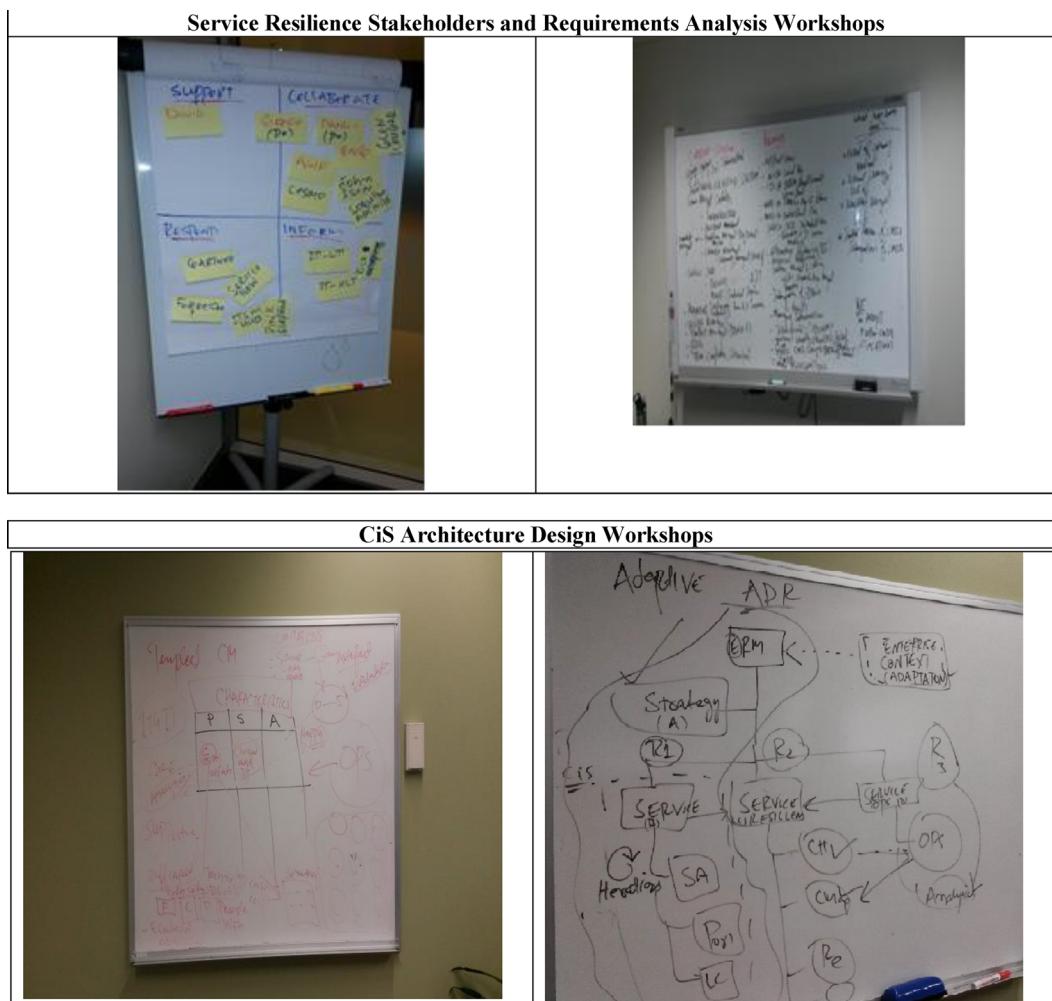


Fig. A1. Examples of Analysis and Design Workshops Photos.

Appendix B

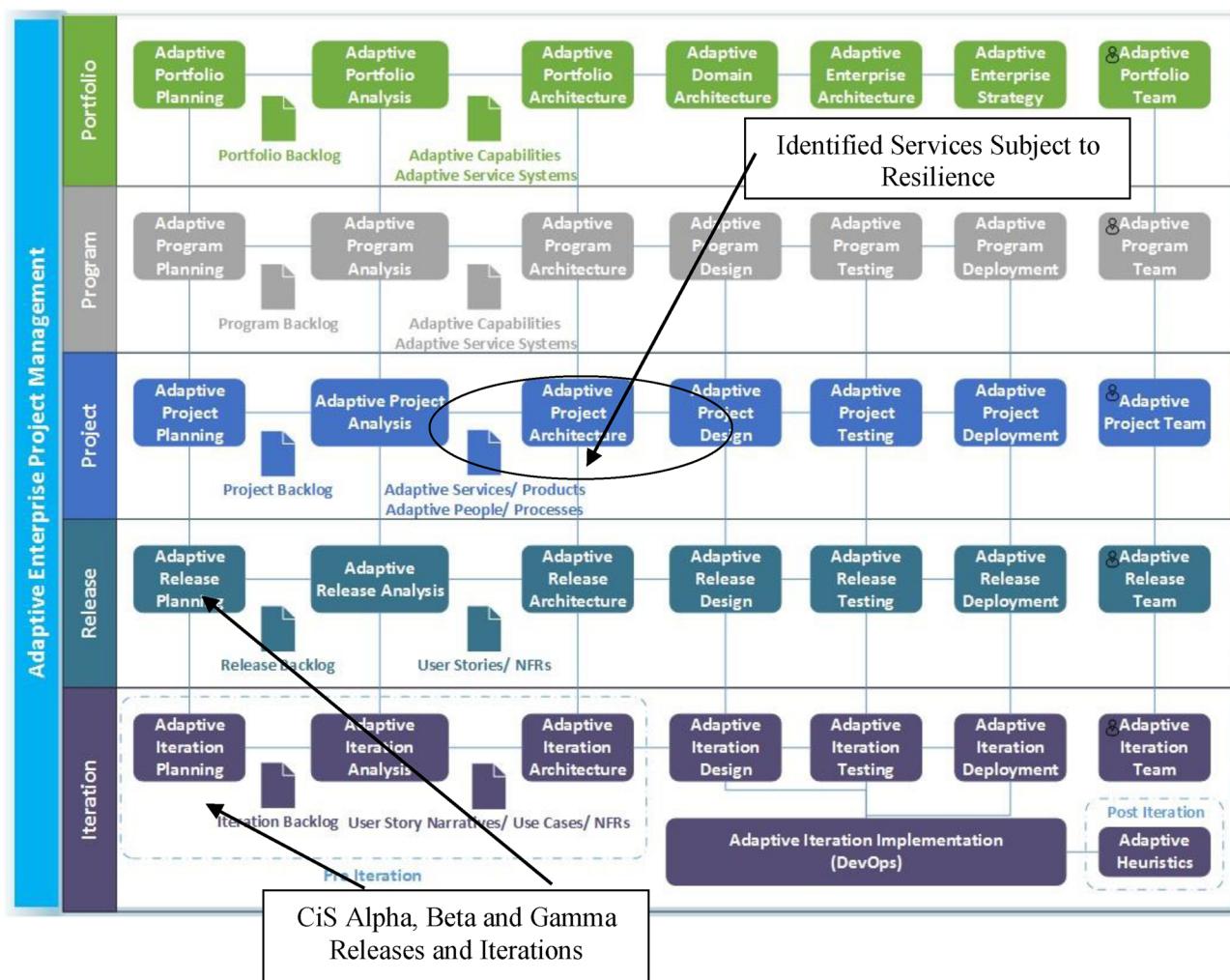


Fig. B1. The Adaptive Enterprise Project Management Capability Reference Model.

References

- [1] R. Schneider, J. Sledge, The Future of Financial Services: Recommendations for Asset Building, Centre for Financial Services Innovation, 2011, <http://www.cfsinnovation.com/content/future-financial-services-0>.
- [2] J. Mylonakis, "A review of banking institutions' transformation in balkan transition economies: 1990-2000", Int. J. Financ. Serv. Manage. 1/2 (2) (2007) 100–117.
- [3] IBM, "Business Resilience: the Best Defense Is a Good Offense: Develop a Best Practices Strategy Using a Tiered Approach", (2009) http://www.ibm.com/smarterplanet/global/files/us_en_us_security_resiliency_buw03008usen.pdf.
- [4] A.Q. Gill, "Social architecture considerations in assessing social media for emergency information management applications", Aust. J. Emerg. Manage. 30 (January (1)) (2015) 17–21.
- [5] R.A. Caralli, J.H. Allen, D.W. White, CERT Resilience Management Model, SEI, 2011.
- [6] J. Lundberg, B.J. Johansson, Systemic resilience model, Reliab. Eng. Syst. Saf. 141 (2015) 22–32.
- [7] D.D. Woods, Four concepts for resilience and the implications for the future of resilience engineering, Reliab. Eng. Syst. Saf. 141 (2015) 5–9.
- [8] A.Q. Gill, E. Chew, G. Bird, D. Kricker, An agile service resilience architecture capability: financial services case study, Business Informatics (CBI), 2015 IEEE 17th Conference on 1 (2015) 209–216.
- [9] K. Furuta, T. Kanno, Key issues in service systems resilience. Proceedings of 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), (2013).
- [10] L. Guddeemi, Why 85% Of Companies Fail At Creating A Configuration Management Database (CMDB), (2014) <http://www.forbes.com/sites/sungardas/2014/10/22/>
- [11] T. Isenbe, Top 10 Reasons a CMDB Implementation Fails, (2011) <http://www.businessservicemanagementhub.com/2011/01/18/top-10-reasons-a-cmdb-implementation-fails/>.
- [12] N. Papas, R. O'Keefe, P. Seltsikas, The action research vs design science debate: reflections from an intervention in eGovernment, Eur. J. Inf. Syst. 21:2 (2012) 147–159.
- [13] A.M. Olivos, From Individual to Organizational Resilience, A Case Study Review, (2014) Master Thesis.
- [14] C.S. Holling, Resilience and stability of ecological systems, Annu. Rev. Ecol. Syst. 4 (1973) 1–23.
- [15] A.W. Riggi, T.A. Saurin, P. Wachs, A systematic literature review of resilience engineering: research areas and a research agenda proposal, Reliab. Eng. Syst. Saf. 141 (2015) 142–152.
- [16] L.H. Gunderson, C.S. Holling, Panarchy: Understanding Transformations in Human and Natural Systems, Island Press, Washington, D.C., 2002.
- [17] B.H. Walker, L.H. Gunderson, A.P. Kinzig, C. Folke, S.R. Carpenter, L. Schultz, A handful of heuristics and some propositions for understanding resilience in social-ecological systems, Ecol. Soc. 11 (1) (2006) 13 <http://www.ecologyandsociety.org/vol11/iss1/art13/>.
- [18] E.A.M. Limnios, T. Mazzarol, A. Ghadouani, S.G. Schilizzi, The resilience architecture framework: four organizational archetypes, Eur. Manage. J. 32 (1) (2014) 104–116.
- [19] L. Shen, L. Tang, A resilience assessment framework for critical infrastructure systems, Proceedings of 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), (2013).
- [20] J. Lundberg, B.J. Johansson, Systemic resilience model, Reliab. Eng. Syst. Saf. 141 (2015) 22–32.

- [21] K. Furuta, T. Kanno, Key issues in service systems resilience, Proceedings of 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W), (2013).
- [22] A.Q. Gill, D. Bunker, P. Seltsikas, Moving Forward: Emerging Themes in Financial Services Technologies' Adoption, *Commun. Assoc. Inf. Syst.* 36 (2015).
- [23] A. Rose, S.Y. Liao, Modeling regional economic resilience to disasters: a computable general equilibrium analysis of water service disruptions, *J. Reg. Sci.* 45 (1) (2005) 75–112.
- [24] O. Erol, M. Mansouri, B. Sausser, A framework for enterprise resilience using service oriented architecture approach, *Systems Conference, 2009 3rd Annual IEEE* (pp. 127–132) (2009).
- [25] M. Stein, O. Henfridsson, S. Purao, M. Rossi, R. Lindgren, Action design research, *Mis Q.* 35 (1) (2011) 37–56.
- [26] M.D. Myers, J.R. Venable, A set of ethical principles for design science research in information systems, *Inf. Manage.* 51 (6) (2014) 801–809.
- [27] R. Baskerville, M.D. Myers, Special issue on action research in information systems: making IS research relevant to practice—foreword, *Mis Q.* 28 (3) (2004) 329–335.
- [28] A.R. Hevner, S.T. March, J. Park, S. Ram, "Design science in information systems research", *Mis Q.* 28 (1) (2004) 75–105.
- [29] J.C. Jones, D.G. Thornley, Conference on Design Methods, Pergamon Press, Oxford, UK, 1963.
- [30] S.T. March, G.F. Smith, Design and natural science research on information technology, *Decis. Support Syst.* 15 (4) (1995) 251–266.
- [31] I. Benbasat, R.W. Zmud, Empirical research in information systems: the practice of relevance, *Mis Q.* (1999) 3–16.
- [32] A. Dennis, Relevance in information systems research, *Commun. AIS* 6 (10) (2001).
- [33] R. Winter, Design science research in Europe, *Eur. J. Inf. Syst.* 17 (5) (2008) 470–475.
- [34] R.B. Gallupe, The tyranny of methodologies in information systems research, *SIGMIS Database* 38 (3) (2007) 20–28.
- [35] W.J. Orlikowski, C.S. Iacono, Research commentary: desperately seeking the 'IT' in IT research—a call to theorizing the IT artifact, *Inf. Syst. Res.* 12 (2) (2001) 121–134.
- [36] J. Iivari, A paradigmatic analysis of information systems as a design science, *Scand. J. Inf. Syst.* 19 (2) (2007) 39–63.
- [37] S. Purao, M. Rossi, M.K. Stein, On integrating action research and design research, *Design Research in Information Systems*, Springer, US, 2010, pp. 179–194.
- [38] S. Gregor, A. Hevner, Positioning and presenting design science research for maximum impact, *Mis Q.* 37 (2) (2013) 337–355.
- [39] N.K. Denzin, Y.S. Lincoln, *Handbook of Qualitative Research*, 2nd ed., (2000).
- [40] A. Qumer, B. Henderson-Sellers, An evaluation of the degree of agility in six agile methods and its applicability for method engineering, *Inf. Softw. Technol.* 50.4 (2008) 280–295.
- [41] D. Gray, *Doing Research in the Real World*, 2nd ed., SAGE Publications Ltd., London, 2009.
- [42] R.L. Baskerville, A.T. Wood-Harper, Diversity in information systems action research methods, *Eur. J. Inf. Syst.* 7 (2) (1998) 90–107.
- [43] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research, *J. Manag. Inf. Syst.* 24 (3) (2008) 45–77.
- [44] M. Mandviwalla, Generating and justifying design theory, *J. Assoc. Inf. Syst.* 16 (5) (2015).
- [45] R. Martin, *The Design of Business: Why Design Thinking Is the Next Competitive Advantage*, third edition, Harvard Business Review Press, 2009.
- [46] A.Q. Gill, "Adaptive Cloud Enterprise Architecture", World Scientific Publishing Co., 2015.
- [47] J.G. Miller, "Living systems", University Press of Colorado, 1995.
- [48] J. Spohrer, S. Vargo, N. Caswell, P. Maglio, "The service system is the basic abstraction of service science", 41st Annual HICSS Conference Proceedings, (2008).
- [49] OMG, Model Driven Architecture. <http://www.omg.org/mda/>.
- [50] The Open Group, TOGAF 9.1. <https://www.opengroup.org/togaf/>.
- [51] itSMF, ITIL. <http://www.itsmf.org.au/?page=ITILInfrastructure>.
- [52] Agile Manifesto, Manifesto for Agile Software Development, 2001, <http://agilemanifesto.org/>.
- [53] D. Tilson, K. Lyytinen, C. Sørensen, Research commentary—digital infrastructures: the missing IS research agenda, *Inf. Syst. Res.* 21 (4) (2010) 748–759.
- [54] O. Hanseth, K. Lyytinen, Design theory for dynamic complexity in information infrastructures: the case of building internet, *J. Inform. Technol.* 25 (1) (2010) 1–19.
- [55] P.A. David, The Beginnings and Prospective Ending of "end-to-end": an Evolutionary Perspective on the Internet's Architecture, Discussion Paper 0502012, EconWPA, Stanford, CA, 2005 <http://ideas.repec.org/p/wpa/wuwpio/0502012.htm>.
- [56] D. Tilson, The Interrelationships Between Technical Standards and Industry Structures: Actor-network Based Case Studies of the Mobile Wireless and Television Industries in the U.S. And the UK. Ph.D. Thesis, Case Western Reserve University, Cleveland, 2008 <http://hdl.handle.net/2374.OX/17205>.
- [57] Ola Henfridsson, Bendik Bygstad, The generative mechanisms of digital infrastructure evolution, *MIS Q.* 37 (September (3)) (2013) 907–932 (2013).

Dr. Asif Qumer Gill Asif Gill is a result-oriented academic and practitioner with extensive experience in successfully delivering multi-million dollar projects in various sectors including banking, consulting, education, finance, government, nonprofit, software and telco. He conducts customer-centric practice-oriented applied action-design research that targets the challenges of academia, industry, government and society. His work has appeared in major academic and industry journals including *Information and Management*, *Information Systems*, *Communications of the AIS*, *Information and Software Technology*, *Journal of Systems and Software*, *Cutter IT Journal* and *CIO Magazine*. He is author of the Adaptive Cloud Enterprise Architecture book and inventor of The Gill Framework® for architecting Adaptive Enterprises as Adaptive Enterprise Service Systems. He is often invited and involved as a professional speaker, guest editor, conference chair, organizer and reviewer for a number of national and international academic and industry conferences.

Professor Eng Chew Eng Chew is Professor of Business and IT Strategy at the University of Technology, Sydney (UTS), and part-time Industry Advisor. From 2005–2008, he held the UTS-Gartner Chair of Business and IT Strategy at UTS. He is a former Chief Information Office of SingTel Optus, and has over 25 years of industry experience in IT and Telecommunications in Australia. His achievements include delivery of several hundreds of million Australian dollars of business value through business process re-engineering and organisational transformation. As a natural extension to his industry practice experience, Professor Chew's research interest is on information and technology management and leadership practices, particularly in the context of service innovation.